

Workshop on Wireless Sensing

Proceedings

Sensors Expo & Conference

June 4, 2001

Chicago, Illinois



Kang Lee,
Manufacturing Metrology Division

Jim Gilsinn,
Intelligent Systems Division

Rick Schneeman,
Manufacturing Metrology Division

Hui-Min Huang,
Intelligent Systems Division

TABLE OF CONTENTS

I. Overview of the Wireless Sensing Workshop	4
II. Issues & Discussions	5
III. PowerPoint presentations	
“Introduction to IEEE P1451” Kang Lee, NIST	11
“Implementation of IEEE 1451.1 at NIST” Rick Schneeman, NIST	25
“Bluetooth Overview” Thurston Brooks, 3e Technology International	44
“Wireless Ethernet (802.11) Overview” Jim Gilsinn, NIST	66
“Wireless Interface Options of 1451” Mike Moore, Oak Ridge National Lab	74
“Developing and Executable Specification for Networking Smart Transducer to Bluetooth” Steve Bibyk, Ohio State University	80
“Development of Wireless Sensor Technology for Machine Monitoring” Mark Bocko, Oceana Sensors	96
IV. List of Attendees	105

I. Overview of the Wireless Sensing Workshop

The first Wireless Sensing Workshop was held on June 4, 2001, at the Sensors Expo/Conference at the Rosemont Convention Center in Chicago, IL. The National Institute of Standards and Technology (NIST), SENSORS magazine, Sensors Conference, and Institute of Electrical and Electronics Engineer (IEEE) Instrumentation and Measurement Society's Technical Committee on Sensor Technology (TC-9) cosponsored the workshop. NIST is an agency of the U.S. Department of Commerce's Technology Administration. Its mission is to help increase U.S. industry competitiveness through advanced research, standards, and technology collaboration.

Recently, there has been considerable interest from industry and government in applying wireless technology to sensor-based applications. This is due primarily to the prolific phenomenon of Bluetooth, a wireless technology being developed by a 1200-member industrial consortium. According to earlier Bluetooth industry predictions, a billion Bluetooth wireless devices may be in use all over the world within five years. Bluetooth technology providers indicated that they could provide low cost, seamless integration of wireless devices from home automation to mobile systems, office automation, manufacturing facilities, and field operations. Other technology, such as Ethernet, has become dominant in network communication, and its usage is becoming increasingly popular in manufacturing. Wireless Ethernet has been moving from office automation into other application areas, including home and factory automation. Sensor companies have begun developing and applying these standard interfaces to sensor applications. The US Navy has also expressed interest in wireless sensor connectivity aboard naval vessels to enhance overall system performance, reduce manpower, and increase efficiency.

The Sensor Development and Application Group at NIST has been working with industry and IEEE to establish IEEE 1451, titled A Standard for a Smart Transducer Interface for Sensors and Actuators. In response to the industry's interest in wireless sensing, NIST initiated, cosponsored, and conducted this workshop to explore this level of interest. In addition, state-of-the-art, wireless communication technologies were examined. This workshop provided a good opportunity for representatives from industry, academia, and government to discuss the possibility of a standard for wireless sensing in an open forum. Ninety people attended the workshop to represent the manufacturing, process control, aerospace, home automation, automotive, and government sectors. The ratio of attendees was approximately 4/2/1 for users/sensor vendors/network vendors, respectively.

The workshop opened with an overview of the IEEE 1451 standard. NIST's reference implementation of the IEEE 1451.1 smart transducer information model and the investigation of interfacing the 1451.1 model to the wireless world were discussed. Then various wireless technologies such as the wireless Ethernet standard (IEEE 802.11x) and Bluetooth were presented in detail. Following that, hardware and software tools that could help speed up wireless application development, as well as the application of wireless Bluetooth technology for sensors, were presented. One presentation proposed a wireless sensor interface standard, a potential IEEE P1451.5, using the IEEE 802 as a guideline for managing the IEEE 1451 framework.

After briefing the attendees on various communication interface standards, an open forum discussion began. Attendees were encouraged to provide input regarding their needs and general

requirements for a wireless sensor communication interface. The results of the discussions are presented in Section III: Issues and Discussions.

The open forum appeared to be successful in determining the appropriateness of various wireless communication technologies for sensor interfacing. It has also begun the dialogue in assessing the general wireless requirements of sensor manufacturers and users. By the request of the attendees, a follow-up wireless sensor workshop has been scheduled for October 4, 2001, at the next Sensors Expo/Conference in Philadelphia, PA. Together we will further examine other technologies and begin pursuing the IEEE procedure for organizing a working group for developing a wireless communication interface standard for sensors.

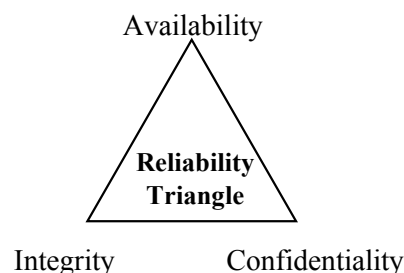
III. Issues & Discussions:

Why use wireless at all?

Some attendees at the conference questioned whether wireless communications for sensors should even be considered at all. For applications such as high-speed process control, wireless might not be the best solution. In those cases, wired, dedicated sensors may be the best solution. However, in many cases where processes exist over a large facility or are low enough speed, wireless may be a best-fit solution since the cost of cabling can be exorbitant when compared to the cost of the sensor itself.

What is data reliability and how does it affect wireless sensors?

Data reliability was the largest issue raised during the workshop. Data reliability depends on three factors: availability of the wireless signal, integrity of the data message, and confidentiality of the data message. These three factors were brought up in many different forms all throughout the presentations and discussions at the workshop.



Availability of the wireless signal is the physical side of the reliability triangle. Some issues related to the availability are range, interference, and data throughput. *"Range is everything!"* By increasing the range of a signal, the signal is less prone to multi-path effects and small signal interference. *"The good news about the ISM band is you don't need a license. The bad news is nobody else does either."* Many of the wireless sensors available or being developed use the Industrial, Scientific and Medical (ISM) bands for their transmission. This allows them the freedom of not relying on the Federal Communications Commission (FCC) for regulation of signal. However, it also limits power and sole proprietorship for a particular bandwidth. Many

of the data transmission algorithms used today have compensation built-in to overcome interference in some way. Some of the standards developed for wireless communications allow multiple data throughput speeds to be used depending on the distance between the transmitter and receiver. IEEE 802.11b allows the throughput to step down from 11 to 5.5 to 1 Mbps if the two systems begin to move outside the recommended operating range.

Data integrity and confidentiality are the software components of the reliability triangle. In order for modern wireless communications to send reliable data from one place to another, the integrity of that data needs to be checked. Data integrity basically establishes that the bits you send get to the receiver and that they receive the correct bits of data. Many of the wireless communication algorithms developed today have some sort of data integrity or error correction checks built-in, such as cyclical redundancy checks (CRC). More complicated algorithms use some sort of forward error correction that allows for correcting errors without requesting the signal be sent again. These algorithms can increase the effective bandwidth of a communication link, since there are fewer requests for re-transmission made by the devices on the network. The IEEE 1451.2 specification for a Transducer Electronic Data Sheet (TEDS) has error correction built-in using a checksum, but requires the user to include their own error detection code if they want a more complicated algorithm.

Data confidentiality relates to the security of the signal being sent from one system to another. This can be an important feature of some wireless communication systems, since companies worry about having their data co-opted by a competitor. As more and more data is stored and moved digitally, industrial espionage has become an ever more present threat to companies.

What are users particular bandwidth requirements?

An informal survey of the bandwidth requirements for workshop attendees was conducted. The results are as follows.

Bandwidth	Interested Parties
<= 300 bps	63%
300 bps – 50 kbps	25%
50 kbps – 250 kbps	3%
250 kbps – 1.5 Mbps	6%
> 1.5 Mbps	3%

Where should the wireless communications be located?

Many of the attendees at the workshop were of the opinion that the wireless communications should not be included directly on the sensors for cost reasons as seen in Figure 1A. Smart Transducer Interface Modules (STIMs) as defined in IEEE 1451.2 consist of sensors and/or actuators, signal conditioning circuit and digital data output. The attendees decided that it would be better to have sensors attached to some sort of wireless NCAP node as shown in Figure 1B, which then communicate to a wired network via a gateway. Network Capable Application Processor (NCAP) is defined in the IEEE 1451.1 as a sensor network node.

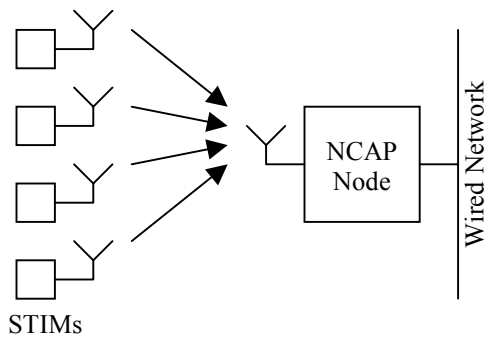


Figure 1A. Wireless STIMs

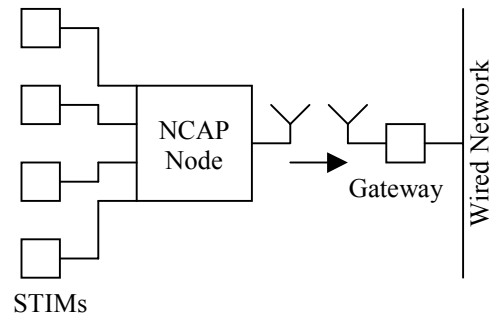


Figure 1B. Wireless NCAP Node

How many devices per node do users require?

NCAP nodes allow multiple sensors to be attached to the network using one common point of access. These allow for the communication portion of the sensor network to be taken out of the sensors themselves and distributed between multiple sensors via a separate piece of hardware. An informal survey was taken of the numbers of sensors for each node. The results are as follows.

Sensors/Node	Interested Parties
8	26%
32	53%
256	21%

How does wireless transmission power affect battery lifetime and safety?

Another issue raised at the workshop had to do with the power requirements of some of the wireless communication standards. Some of them require “high” power (100 mW), which may not be available in small battery operated sensors. The expected battery life of some of the planned sensor needs to be on the order of years, which limits the power consumption allowed for the communication system. Replacing batteries in some machines may not be as easy as it would be for a typical cell phone.

There are also issues of the intrinsic safety of such “high” power communications when used in hazardous environments. The Occupational Safety and Health Administration (OSHA) has restricted the use of wireless Ethernet in chemical plants and refineries for the lack of safety certifications. Wireless communication is also not allowed for fail-safe systems such as emergency stop systems where lives are at stake.

How can wireless sensors benefit from the “hot” wireless technology?

During the workshop presentations, the topic of cost was brought up related to both the basic chip sets for and devices using different wireless communication standards. Although the Bluetooth and 802.11b standards may not be the best fit for all applications in the sensor community, the fact that devices using the chip sets based on these standards are developed in

mass quantities for the mobile computing community means that the chip sets will come down in price due to economy of scale. With custom application specific integrated circuit (ASIC) chips for wireless communications, the best algorithms for a particular application could be implemented, however, the device would cost considerably more due to the development cost of the chip set itself. Although the IEEE 1451 standard is being developed for most applications, it should not eliminate such standards as Bluetooth and 802.11b due to their enormous backing and potential in wireless communications.

Even though the production chip sets for a particular wireless communication standard may be inexpensive, the cost for the development system may be thousands of dollars. This cost could easily be amortized if a company plans to sell thousands of units, but if it plans to be a custom design house, it may be prohibitively expensive to make up for the startup costs for these standards.

How can sensors be reconfigured in place?

Some devices may require different communication speeds for different tasks. Configuring the device initially or upgrading the device's software may require a high-speed 2-way link that allows for a large amount of data to be transmitted all at once. Once the device starts operating normally, it may only require a small amount of bandwidth, especially if the sensor has some intelligence built-in.

What if more than one system needs the sensor data simultaneously?

The question about whether the communication should be broadcasted or targeted came up. Broadcast communications allow one producer to broadcast its information to multiple receivers without knowing who they are or how many receivers are listening. Although this may be good in some cases, it is not appropriate in all cases. Many networks do not allow this type of communication, so the issue will need to be addressed. As of yet, it has not been brought up in the IEEE 1451 standard.

Can multiple wireless sensors synchronize their data at high speed?

Event synchronization may be difficult at less than 1 ms using standard wireless communications. Some standards have built-in time synchronizing capabilities due to the fact that they must do something every so often in order to stay as part of the network. Bluetooth devices, for example, must hop frequencies every 625 μ s, making synchronizing at 1 ms very easy. To go much farther down would take a special timing chip set on the device that is capable of special synchronization in order to coordinate multiple devices. A proposed IEEE P1588, titled A Standard for Precise Clock Synchronization in Networked Measurement and Control Systems, is being developed in the IEEE Instrumentation and Measurement Society's Technical Committee on Sensor Technology to address this kind of issue.

Can the NCAP and STIM Be Combined?

As specified in the standard, IEEE 1451.2 defined a physical 10-wire connection between the NCAP and STIM. This was done to allow the plug and play of sensors and networks from different manufacturers. However, the standard does not preclude the NCAP and STIM to be co-located inside the same chip. In such configuration, the interface between the NCAP and STIM

is not exposed for consideration of plug and play, therefore the 10-wire interface is not important or necessary.

Can the ISM bands support the extra users?

The FCC has certain bandwidths that it has declared open to unlicensed equipment as long as that equipment stays within certain power requirements. These bandwidths are called the ISM bands, and many wireless communication devices use these bands. Here is a listing of some of the ISM bands and devices that use those bands:

- 900 MHz Cell phones, portable phones, home electronics, spread spectrum communications
- 2.4 GHz Portable phones, spread spectrum communications
- 5 GHz Satellite communications

Topics for Further Discussion:

Are multiple versions of the IEEE 1451 wireless standard needed?

It may be necessary to look into multiple flavors of a wireless IEEE 1451. There seems to be enough differences between the low and high-speed sensor communities that multiple standards within the IEEE 1451 framework may be necessary to meet the needs of the two worlds, unless there is a way to define the specification to accommodate the requirements of both groups. This can be a way to break out discussions of power consumption as well, since the high-speed community may not have the same requirements for power consumption as the low-speed community.

Although there were some negative comments, many workshop attendees seemed open to the idea of creating a wireless version of IEEE 1451. A general comment was that the IEEE 1451 committee had done a very good job developing a standard for smart sensors. Enough interest had been developed in the industry that more manufacturers and users were looking into the standard for their particular applications.

Should there be another workshop organized?

There are multiple upcoming conferences, and it would be good to organize a follow-up workshop at one of these. The possible candidates include:

- September 10-13, 2001 ISA 2001, Houston, TX
- October 2-4, 2001 Sensors Expo, Philadelphia, PA
- November 5-7, 2001 SIcon 2001, Chicago, IL

The next workshop is scheduled to be held on October 4, 2001 at the Sensors Expo/Conference in Philadelphia, PA. At the follow-up workshop, a strawman should be developed for applying the IEEE 1451 to a particular wireless sensor application in order to help develop the standard further. Also, some tools for developing the standard should be discussed. These tools can be either hardware tools such as very high density layout (VHDL) development packages or software tools such as unified modeling language (UML) and object-oriented design tools.

How will the proceedings for this workshop be distributed?

Information about this workshop and its proceedings will be made available in both electronic and paper form. Electronic forms can be obtained on compact disk (CD) and from the web. The web address for information is <http://ieee1451.nist.gov>.



Wireless Sensing and IEEE 1451

Sensors Conference / Expo 2001
June 4, 2001

Kang Lee

kang.lee@nist.gov

**National Institute of Standards and Technology
United States Department of Commerce**

K. Lee, NIST, 6/4/01



Outline

- Objectives of the Wireless Sensing Workshop
- Introduction / Background
- State of Industry on Sensor Interfaces
- Sensor Interface Standardization-IEEE P1451
- Summary of IEEE P1451
- Interest in Wireless Sensors
- Acknowledgment

Objectives of the Wireless Sensing Workshop

- Review the latest wireless technologies and their applications.
- Provide an open forum for examining and discussing the appropriateness of these and other technologies for use as wireless sensor interfaces.
- Discuss the requirements of wireless sensor communication interfaces.
- Use previous IEEE P1451 model and experience to explore wireless interface standardization.
- Charter a future action plan

K. Lee, NIST, 6/4/01

Introduction / Background

- The National Institute of Standards and Technology (NIST) is part of the Department of Commerce's Technology Administration.
- NIST's Mission: To help increase U.S. industry competitiveness through advanced research, standards, and technology collaboration.
- The Sensor Development and Application Group at NIST has been working with industry and IEEE - establishing the Smart Transducer Interface Standard, IEEE 1451.

K. Lee, NIST, 6/4/01

State of Industry on Sensor Interfaces

- Smart features integrated into sensors and actuators.
- Increasing uses of digital communication and networked configurations for connecting sensors and actuators.
- The trend is moving toward distributed measurement and control, and distributed intelligent sensing architecture.
- Networked sensor technology applied to commercial and consumer applications in
 - process control,
 - industrial automation,
 - automotive,
 - aerospace,
 - to smart buildings and homes
 - etc...

K. Lee, NIST, 6/4/01

State of Industry - cont'd

- Networked sensors are needed
 - *Example:* US Navy needs tens of thousands of networked sensors per vessel to enhance automation because of the reduced-manning program.
 - *Example:* Boeing needs to network hundreds of sensors to monitor and characterize airplane wing performance. Boeing would benefit by using networked sensors and actuators to reduce the amount of wiring.



Picture courtesy IEEE



Pictures courtesy Endevco

K. Lee, NIST, 6/4/01

State of Industry - cont'd

- Barriers:
 - Large number of different networks to support.
 - Significant sensor interface software development for each network.
 - Lack of network software know-how and support by sensor manufacturers.
 - *Lack of a standardized sensor interface.*
 - These hold back the speedy development and adoption of smart sensors.

K. Lee, NIST, 6/4/01

Why Networking Sensors?

Networking sensors provides advantages not readily available with traditional sensors, they:

- significantly lower the total system cost by simplified wiring
- enable self-describing of sensors
- allow multi-variable sensor configuration
- enable time stamp of the measurements
- enable embedded processor or ASIC implementation
- ease software configurability
- enable bi-directional digital communication
- communicate messages in standardized digital format
- provide Internet connectivity, thus global, or *anywhere*, access of “sensor information”

K. Lee, NIST, 6/4/01

Sensor Interface Standardization - IEEE P1451

- A set of standards aimed to simplify transducer (sensor or actuator) connectivity.
- It is developed as:
 - an **open**, industry consensus standard with participation from sensor, measurement, and control network industries, and users.
- It's purposes are to:
 - provide a set of common interfaces for connecting sensors and actuators to **existing** instruments, and control and field networks.
 - provide an easy upgrade path for connecting transducers, instruments, or networks from **any** manufacturer.

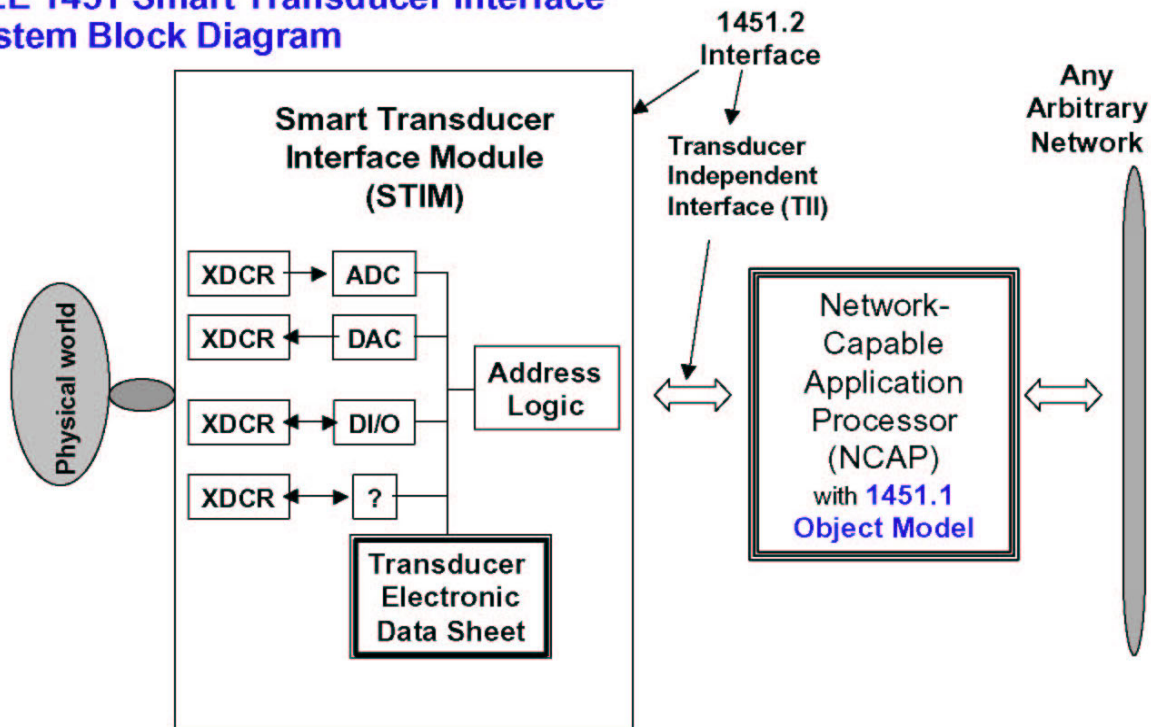
K. Lee, NIST, 6/4/01

What Standards are being developed ?

- **IEEE Std 1451.1-1999**, Network Capable Application Processor (NCAP) Information Model for smart transducers -- ***Published standard.***
- **IEEE Std 1451.2-1997**, Transducer to Microprocessor Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats -- ***Published standard.***
- **IEEE P1451.3**, Digital Communication and Transducer Electronic Data Sheet (TEDS) Formats for Distributed Multidrop Systems -- ***Being developed.***
- **IEEE P1451.4**, Mixed-mode Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats -- ***Being developed.***

K. Lee, NIST, 6/4/01

IEEE 1451 Smart Transducer Interface System Block Diagram



K. Lee, NIST, 6/4/01

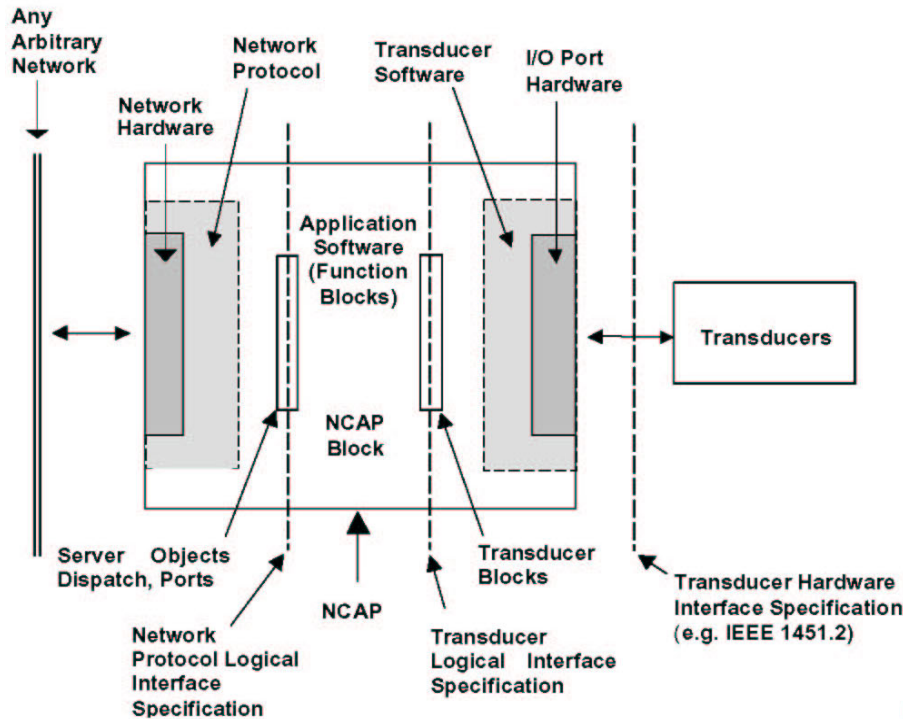
IEEE 1451.1 NCAP Information Model for Smart Transducers

By defining a common object model for the components of a networked smart transducer, together with interface specifications to these components, the Standard provides:

- Network protocol logical interface specification (via Server Object Dispatch and Ports)
 - Software interfaces between application functions in the NCAP and the network in a manner independent of any specific network
- Transducer logical interface specification (via Transducer Block)
 - Software interfaces between application functions in the NCAP and the transducers in a manner independent of any specific transducer driver interface

K. Lee, NIST, 6/4/01

IEEE 1451.1 Networked Smart Transducer Model



K. Lee, NIST, 6/4/01

IEEE 1451.2 Transducer Electronic Data Sheet (TEDS)

- **Meta-TEDS**
 - Data structure related information
 - version number
 - number of implemented channels
 - future extension key
 - ...
 - Identification related information
 - manufacturer's identification
 - model number
 - serial number
 - revision number
 - date code
 - product description
 - ...

K. Lee, NIST, 6/4/01

IEEE 1451.2

Transducer Electronic Data Sheet (TEDS) - cont'd

- **Channel TEDS**

- Transducer related information
 - lower range limit
 - upper range limit
 - physical unit
 - unit warm-up time
 - uncertainty
 - self test key
 - ...
- Data Converter related information
 - channel data model
 - channel data repetitions
 - channel update time
 - channel read setup time
 - channel write setup time
 - data clock frequency
 - channel sampling period
 - trigger accuracy
 - ...

K. Lee, NIST, 6/4/01

IEEE 1451.2

Transducer Electronic Data Sheet (TEDS) - cont'd

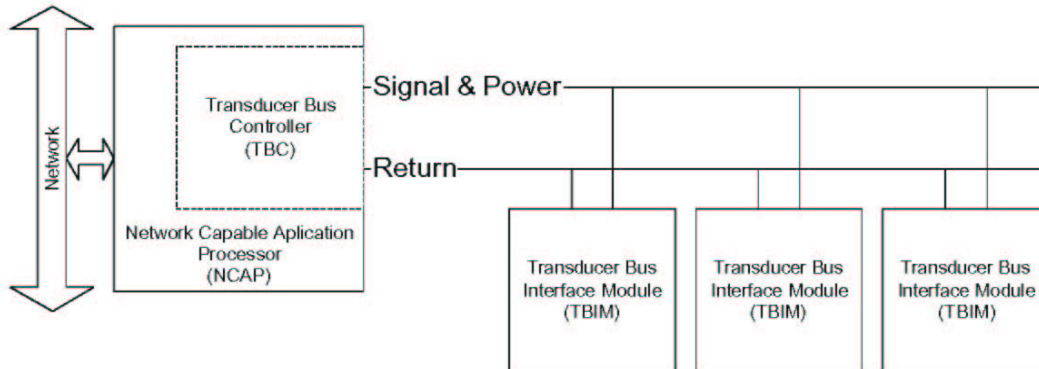
- **Calibration TEDS**

- Data structure related information
 - Calibration TEDS length
- Calibration related information
 - last calibration date-time
 - calibration interval
 - number of correction input channels
 - multinomial coefficient
 -
- Data integrity information
 - checksum for calibration TEDS

K. Lee, NIST, 6/4/01

IEEE P1451.3 Distributed Multidrop System

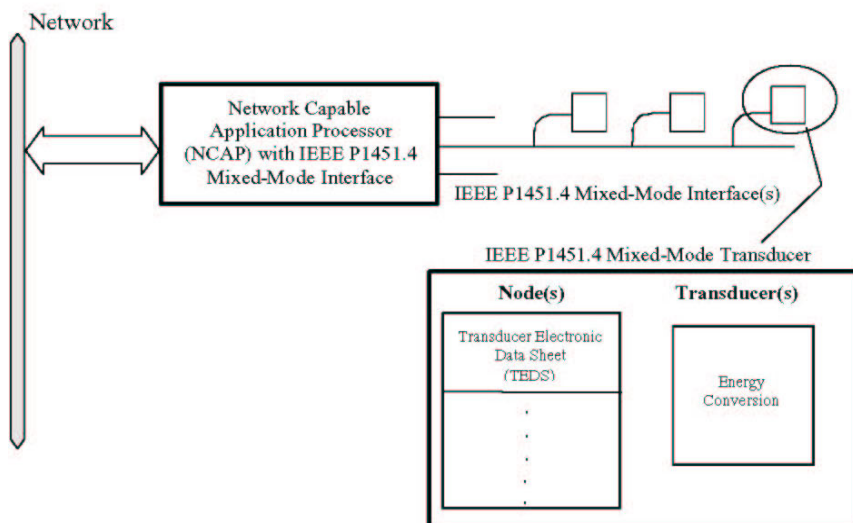
- Synchronously reads large arrays of sensors at high speed in a parallel transducer bus setting.
- Supports sensors with bandwidth requirements to several hundred kilohertz and time correlation requirements in the range of nanoseconds.



K. Lee, NIST, 6/4/01

IEEE P1451.4 Mixed-mode Transducer and Interface

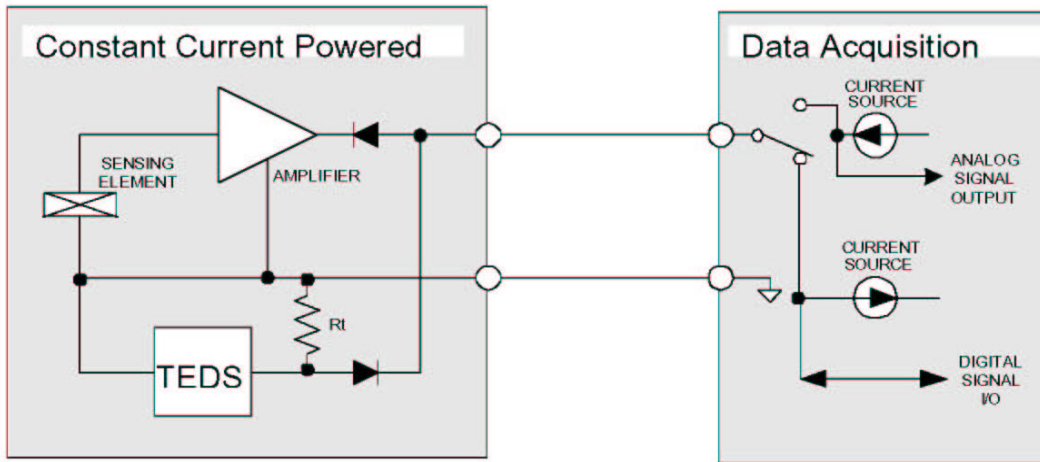
- simple, low-cost connectivity (2-4 wires) of analog sensors with TEDS.
- Support existing or legacy data acquisition systems with TEDS.



K. Lee, NIST, 6/4/01

IEEE P1451.4 Mixed-mode Interface

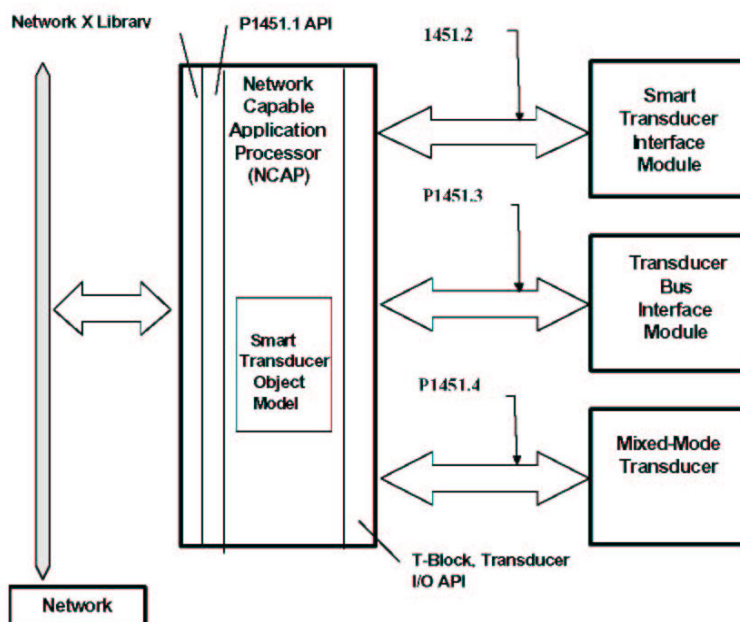
Using Two-Wire Scheme for sending Digital TEDS and Analog Data



K. Lee, NIST, 6/4/01

Summary

IEEE P1451 Family Member Independence

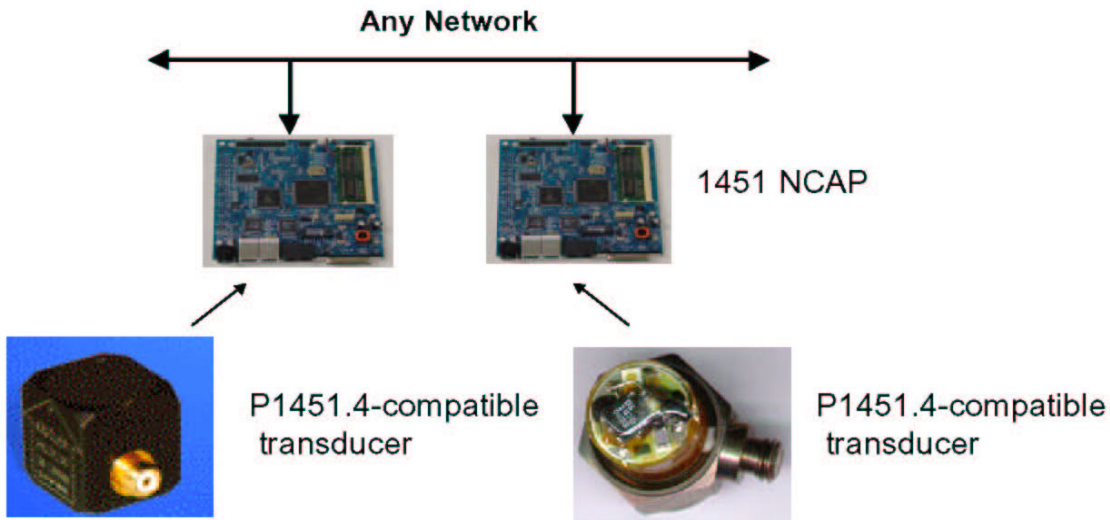


- The proposed standards are being designed to **work with each other**.

- However, each proposed standard can also be **used by itself**, independent of the others.

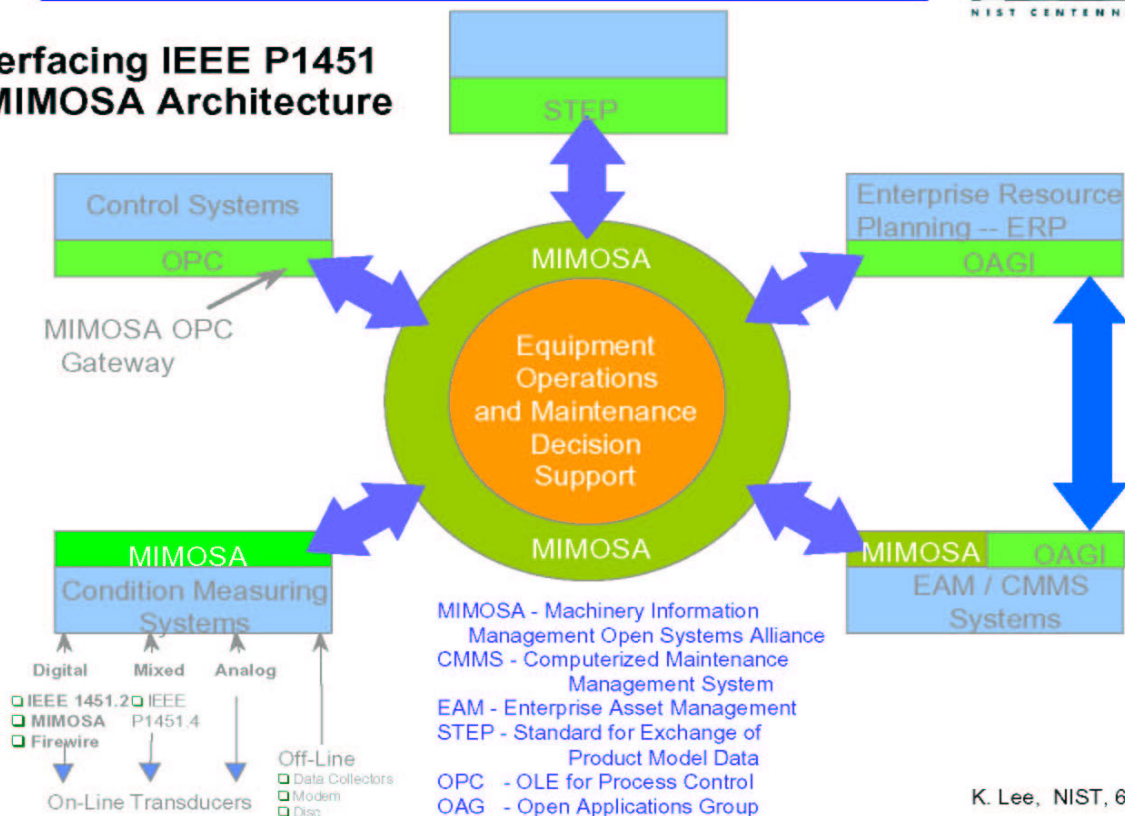
K. Lee, NIST, 6/4/01

IEEE P1451 Enables “Plug and Play” of Transducers to networks

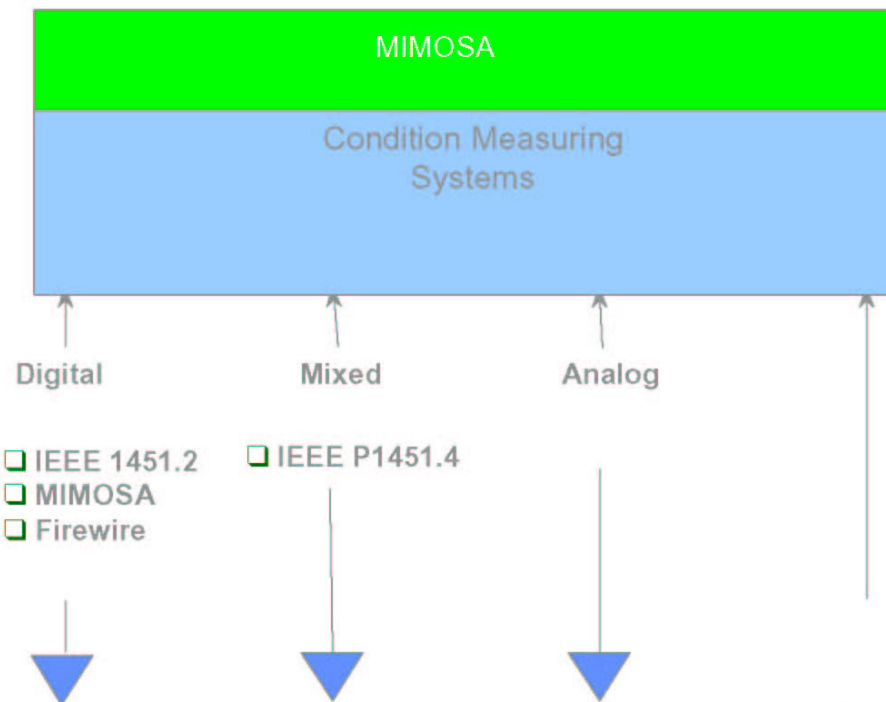


K. Lee, NIST, 6/4/01

Interfacing IEEE P1451 to MIMOSA Architecture



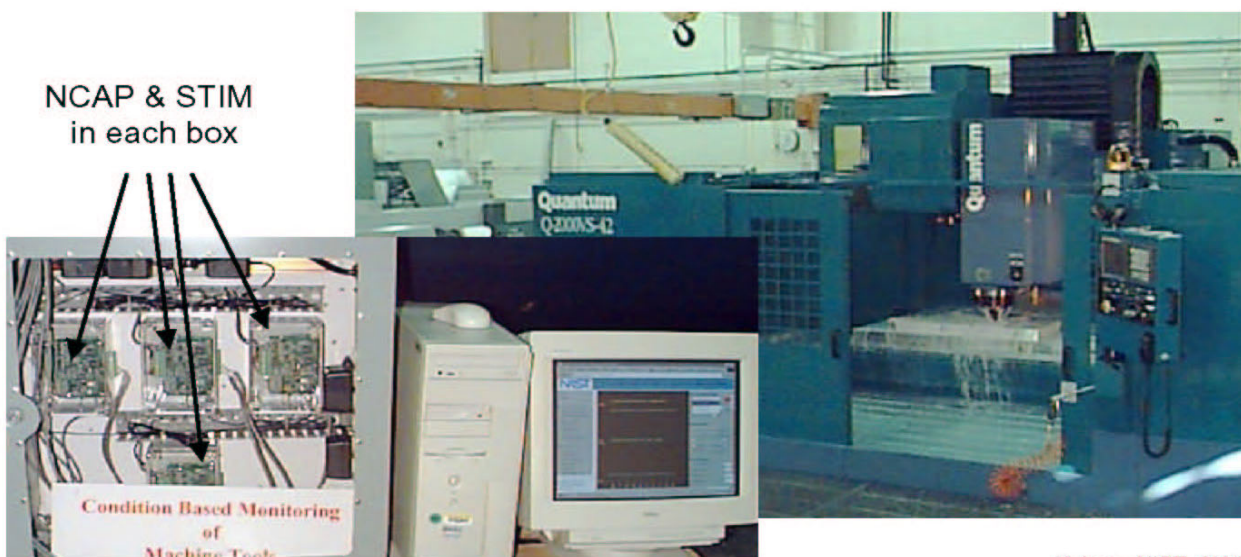
K. Lee, NIST, 6/4/01



K. Lee, NIST, 6/4/01

Machine Condition Monitoring in the Shop

- Temperature sensors monitor spindle motors, bearings, axis drive motors.
- Allow monitoring of sensors over the Internet via any common web browser.



K. Lee, NIST, 6/4/01



K. Lee, NIST, 6/4/01

Wireless Technology is Here

- Traditional proprietary wireless communication protocols could be expensive to implement.
- Wireless capability is being designed into laptop computers, desktop computers, personal data adapters (PDA), and hand-held computers and devices.
- The popular wireless standards are Bluetooth (or IEEE P802.15) and Wireless Ethernet (IEEE 802.11b).
- Wireless communications have revolutionized the cellular phone industry and provide low-cost electronics for a mass market. This could provide a low-cost base for possible wireless sensor applications.

K. Lee, NIST, 6/4/01

Interest on Wireless Sensors in Industry

- Machinery condition-based monitoring is one of many examples that can use wireless sensor technology,
 - quick installation and thus reduced installation cost.
 - improve ability to easily and quickly re-configure the data acquisition and control system.
 - ability to connect measurement data to the Internet.
- Industrial Automation Study Group of the Bluetooth Special Interest Group (SIG) is in full action pursuing the establishment of a working group.
- *Question: Can we use existing wireless technologies for sensor connectivity in a networked environment ?*

K. Lee, NIST, 6/4/01

Acknowledgment

- Thanks to Sensors Magazine, IEEE I&M Society and NIST for their support and co-sponsorship of the workshop.

K. Lee, NIST, 6/4/01

Implementing IEEE 1451.1 in a Wireless Environment

Rick Schneeman, Computer Scientist
rschneeman@nist.gov

US Department of Commerce
National Institute of Standards and Technology (NIST)
Gaithersburg, Maryland 20899 USA

Introduction



- Who we are: NIST mission is to help increase US industry competitiveness through advanced research, standards, and technology collaboration
- Member of the Sensor Development and Application Group (SDAG) within the Manufacturing Engineering Laboratory (MEL) at NIST
- Member of the Working Group on the IEEE Standard for a Smart Transducer Interface for Sensors and Actuators — Network Capable Application Processor (NCAP) Information Model, or IEEE 1451.1 (“dot1”)

Topics of Discussion



- Part 1: Provide a brief object-based overview of the IEEE 1451.1 components, services, and block classes
- Part 2: Discuss the technical and architectural solutions for the development and deployment of the NIST IEEE 1451.1 reference implementation
- Part 3: Illustrate the use of IEEE 1451.1 in an example application using both the NIST C++ and Java reference implementations
- Part 4: Describe an IEEE 802.11b (11Mbps) wireless environment used for application testing and demonstration

Implementing IEEE 1451.1 in a Wireless Environment

IEEE 1451 Overview/Goals



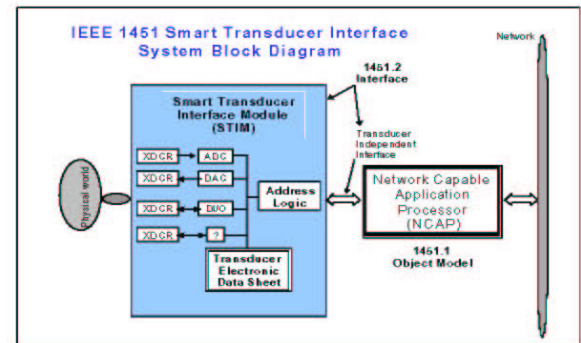
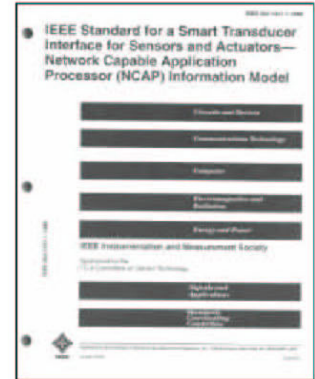
- Provide standardized communication interfaces for smart transducers, both sensors and actuators. In the form of a standard hardware and software definition/specification.
- Simplify the connectivity and maintenance of transducers to device networks through such mechanisms as common Transducer Electronic Data Sheet (TEDS) and standardized Application Programming Interfaces (API)
- Allow plug-and-play with 1451 compatible transducers among different devices using multiple control networks
- Give sensor manufacturers, system integrators, and end-users the ability to support multiple networks and transducer families in a cost effective way

Implementing IEEE 1451.1 in a Wireless Environment

Part 1: IEEE 1451.1 Overview/Goals



- *“The specifications provide a comprehensive data model for the factory floor, and a simple application framework to build interoperable distributed applications...”* Dr. Jay Warrior, Agilent Technologies, Chair IEEE 1451.1 WG
- In general, IEEE 1451.1 accomplishes this by providing:
 - ◆ Transducer application portability (software reuse)
 - ◆ Plug-and-play software capabilities (components)
 - ◆ Network independence (network abstraction layer)
- The standard specifies these capabilities by defining software interfaces for:
 - ◆ Application functions in the NCAP that interact with the network that are independent of any network
 - ◆ Application functions in the NCAP that interact with the transducers that are independent of any specific transducer driver interface



Implementing IEEE 1451.1 in a Wireless Environment

5

IEEE 1451.1 Overview/Goals (Cont.)



- IEEE 1451.1 software architecture is defined using three different models or views of the transducer device environment:
 - ◆ An Object Model, defines transducer device specific abstract objects – or, classes with attributes, methods, and state behavior
 - ◆ A Data Model, defines information encoding rules for transmitting information across both local and remote object interfaces
 - ◆ A Network Communication Model, supports a client/server and publish/subscribe paradigm for communicating information between NCAPs

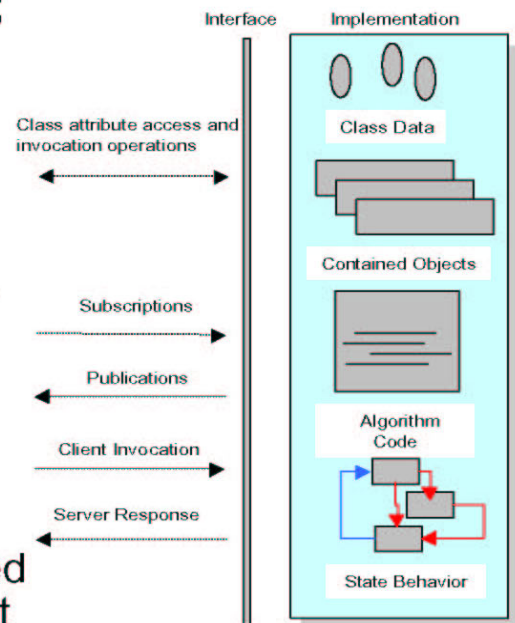
Implementing IEEE 1451.1 in a Wireless Environment

6

Inside an IEEE 1451.1 Object



- IEEE 1451.1 objects look similar to other object-oriented class definitions; they are comprised of instance data, other classes, code for implementing internal operations or methods, and state behavior machines
- Defines services and interfaces required for distributed smart devices (i.e., object discovery, invocation, synchronization) via attribute access and operations
- Specifies Object interfaces and behavior using the Object Model
- The Object Model is formally described using the implementation independent Interface Description Language (IDL)



Implementing IEEE 1451.1 in a Wireless Environment

7

Types of IEEE 1451.1 Classes



- Four Object classes are found in an IEEE 1451.1 system:
 - ◆ **Block Classes** (building blocks of the system)
 - ★ NCAP Block (network communication and configuration)
 - ★ Function Block (application-specific functionality)
 - ★ Transducer Block (transducer device driver interface w/app)
 - ◆ **Component Classes** (common application constructs)
 - ★ Parameter (contains structured information, network variables)
 - ★ Action (time-based system state altering activity)
 - ★ File (supports downloading new code to device)
 - ★ Component Group (addressing collections of related entities)
 - ◆ **Service Classes** (system and network services)
 - ★ Client Ports (implements client-side communication endpoint)
 - ★ Publisher Ports (implements publishing endpoint)
 - ★ Subscriber Ports (implements subscription endpoint)
 - ★ Mutex/Condition Service (provides application/NCAP synch)
 - ◆ **Non-IEEE 1451.1 Classes**

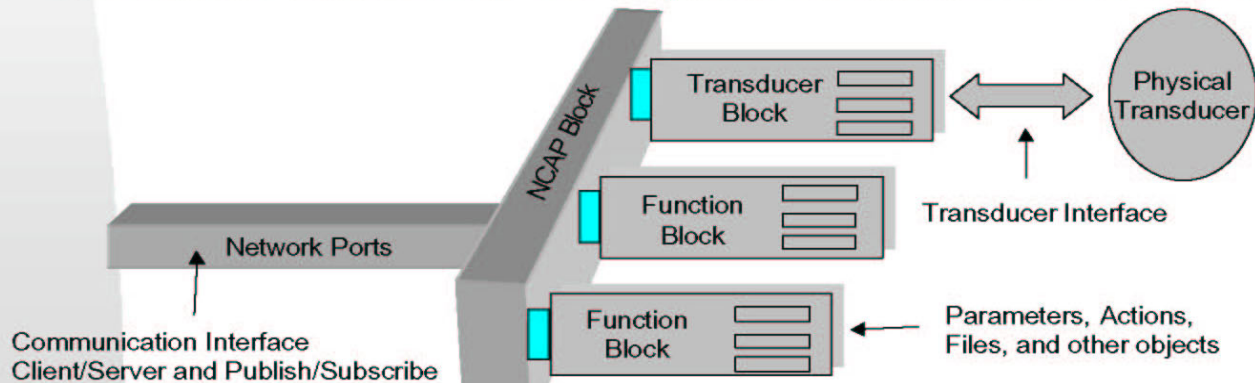
Implementing IEEE 1451.1 in a Wireless Environment

8

Conceptual View of an IEEE 1451.1 NCAP



- Uses a “backplane” or “card cage” concept
- NCAP centralizes and “glues” all the system and communications facilities together
- Network communication viewed through the NCAP as ports
- Function block application code is “plugged” in as needed
- Transducer blocks map the physical transducer to the NCAP



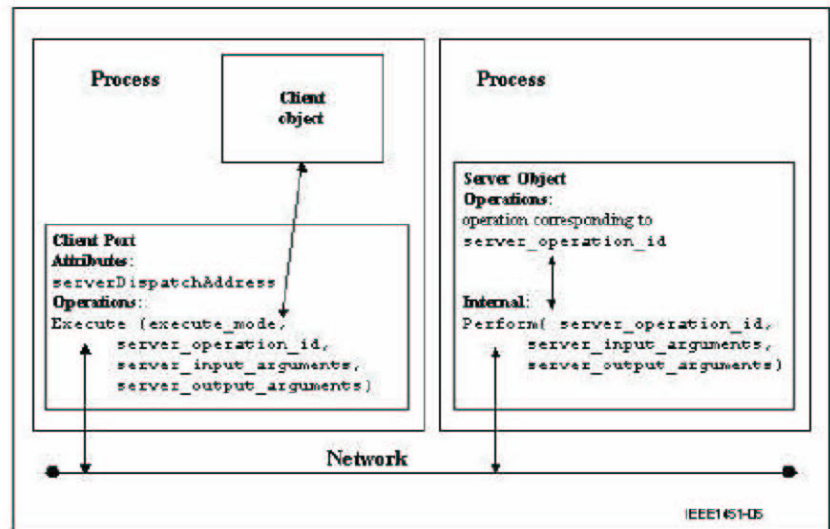
Implementing IEEE 1451.1 in a Wireless Environment

9

IEEE 1451.1 Communication Model



- Provides two styles of inter-NCAP communication
- Client/Server: A tightly coupled, point-to-point model for one-to-one communication scenarios – typically used for configuration, attribute accessors, and operation invocations.



IEEE1451-05

- Client objects “Execute()” or invoke operations over the network against a Server NCAP. Server NCAP objects “Perform()” the operation based on the ID and return the results to the client.

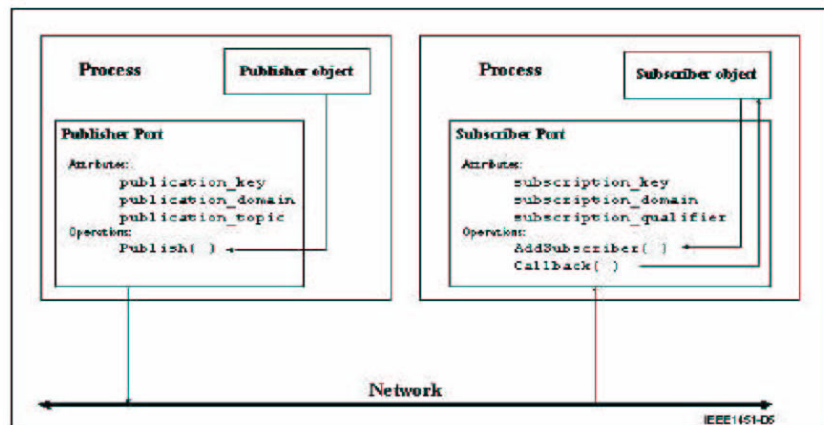
Implementing IEEE 1451.1 in a Wireless Environment

10

IEEE 1451.1 Communication Model (Cont.)



- Publish/Subscribe: A loosely coupled, model for many-to-many and one-to-many communication scenarios – typically used for broadcasting measurement data and configuration management (i.e., node or NCAP discovery) information



- The Publisher is the sending object, it invokes “Publish()” method and does not need to be aware of any receiving objects. Subscribers issue AddSubscriber() method to register interest in something on that subscription.

Implementing IEEE 1451.1 in a Wireless Environment

11

Part 2: Implementing IEEE 1451.1



- An IEEE 1451.1 C++ Reference Implementation provides a concrete representation of the abstract Smart Transducer Information Model (IEEE Std 1451.1-1999, Dated 18 April 2000). The NIST implementation is called “1451.1 Lite”, as it is a subset of the complete specification.
- A subset of the IEEE 1451.1 implementation has also been developed in Java to provide an architecture neutral NCAP configuration tool.
- The C++ implementation uses the open-source Adaptive Communication Environment (ACE) from the Washington University at St. Louis.

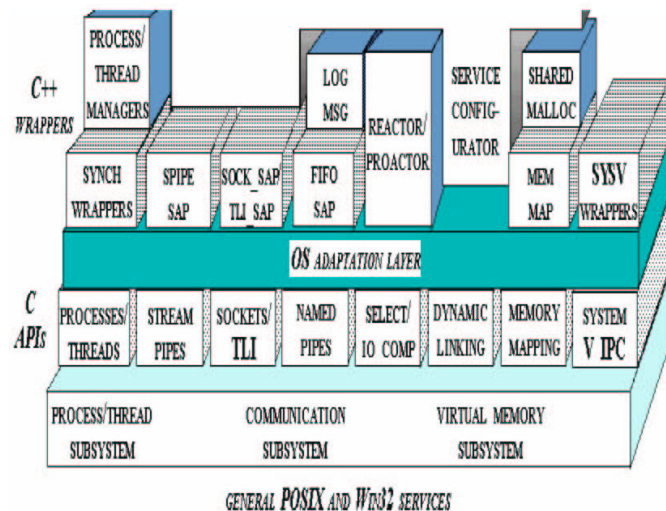
Implementing IEEE 1451.1 in a Wireless Environment

12

Using ACE Framework for IEEE 1451.1



- ACE is an object-oriented framework for implementing portable real-time communication software patterns in C++.
- All core distribution, concurrency, and communication patterns that underlie the NIST developed IEEE 1451.1 implementation are derived from the ACE library.
- Provides an object-oriented abstraction of operating system services for real-time network and communications support.
- Presently used by major communication companies including: Hughes, Lucent, Ericsson, Siemens, and Boeing among others



Implementing IEEE 1451.1 in a Wireless Environment

13

ACE Framework Components



- The Adaptive Communication Environment (ACE) is an object-oriented framework that implements core concurrency and distribution patterns for real-time communication software. ACE includes the following components:
 - ★ Concurrency and Synchronization
 - ★ Interprocess communication (IPC)
 - ★ Memory Management, Timers and Signals
 - ★ File System management
 - ★ Thread Management
 - ★ Event demultiplexing and handler dispatching
 - ★ Connection establishment and service initialization
 - ★ Static and dynamic configuration and reconfiguration of software
 - ★ Distributed communication services –naming, logging, time synchronization, event routing and network locking. etc.
- Notice that the ACE components mimic the object requirements found in the IEEE 1451.1 specification; therefore, the mapping was clear but cumbersome because of the specificity of the standard.

Implementing IEEE 1451.1 in a Wireless Environment

14

Key ACE Architectural Areas Leveraged



- ACE uses well-established object-oriented “patterns”, or common design elements, including:
 - ◆ Reactor pattern (efficient event de-multiplexing and dispatching)
 - ◆ Active Object (multi-threaded execution object)
 - ◆ Activation Queues (decouples method invocation/execution)
 - ◆ Method Objects (queue able objects for execution of commands)
 - ◆ Future Objects (resultant objects of method object execution)
- All of these patterns have concrete representations in ACE, such as ACE_Tasks as an Active Object, etc.
- The low-level C++ TCP/IP socket “wrapper” routines were not used because this would mitigate using the advanced features of the object-oriented framework

Implementing IEEE 1451.1 in a Wireless Environment

15

Key ACE Architectural Areas Leveraged



- IEEE 1451.1 “Entity” class forms the base class for all network and block services in the standard. The “Entity” class inherits the ACE “service handler” class, which integrates event-driven “Active” object patterns with TCP/IP communication endpoints and synchronization.
- ACE “service handler” interface supports real-time event-driven input/output on multicast/unicast TCP/IP sockets, providing an efficient networking scheme.
- The IEEE 1451.1 standard specifies a data encoding sequence for passing parameters between objects. An on-the-wire network-based data representation for marshaling these parameters is required for each control network used. For TCP/IP networks, NIST used a CORBA compliant implementation of the Common Data Representation (CDR) library found in ACE.

Implementing IEEE 1451.1 in a Wireless Environment

16

Key ACE Architectural Areas Leveraged



- Internally, the NIST IEEE 1451.1 code uses the CDR base types as well. This reduces the amount of encoding and marshaling overhead needed from converting base types to the CDR types during marshaling activities.

- Blocks and Ports use ACE “Tasks” for their multithreaded state machine behavior. Each Block/Port class implementation provides a virtualized svc() routine to support each blocks defined state machine.

```
// enter state machine of the fblock
int tempFBlock::svc( void )
{
    while (1)
    {
        // get current state of underlying Block
        ret = IEEE1451_Block::GetBlockMajorState(hms);
        switch (hms)
        {
            case BL_UNINITIALIZED: break;
            case BL_INACTIVE: break;
            // if the block is in the active state, then
            case BL_ACTIVE:
                // get current state of underlying FBlock
                ret = IEEE1451_FunctionBlock::GetFunctionBlockState(fbs);
                // enter the fblock substate state machine
                switch (fbs)
                {
                    case FB_STOPPED: break;
                    case FB_IDLE: break;
                    case FB_RUNNING: break;
                    case FB_RESERVED: break;
                }
                break;
            case BL_RESERVED: break;
            default: break;
        }
    }
    // compiler issue
    return 0;
}
```

Implementing IEEE 1451.1 in a Wireless Environment

17

Progress to Date



- Majority of the design in place (using inheritance and composition of ACE services with 1451.1 code base)
- 70-75% of IEEE 1451.1 C++ code complete, all IEEE 1451.1 methods have placeholders, network encoding still under development
- Core subset (.1 Lite) nearly complete, consists of a shared library under Linux/FreeBSD & VxWorks and a dynamic link library (DLL) under WIN32/NT
- Because the ACE framework is being used, a single source code-base that is 100% portable to Linux/FreeBSD, NT, VxWorks, and other POSIX or WIN32-based operating systems has been developed
- At this time, there is minimal access to the underlying STIM or microprocessor hardware via device drivers, (most I/O such as temperature, pressure, or actuator data has been provided by the NCAP via simulation)
- Java implementation consist of most blocks and communication ports. A CDR encoding from the Zen Project at University of California, Irvine has been integrated and is interoperable with C++ version.

Implementing IEEE 1451.1 in a Wireless Environment

18

Lessons Learned on Implementation



- The standards' low-level attempt to redefine the RPC (Remote Procedure Call) mechanisms for network communication preclude and limit the implementer's ability to optimize the amount and types of communication patterns that can be used in normal real-time object-oriented network communication software design.
- Many object-oriented patterns are disallowed because of arbitrary inheritance chain and class partitioning.
- Partitioning forces the designers and implementers of the software to create overloaded and convoluted classes in order to pigeon hole the standard into certain middleware and object-oriented frameworks.

Implementing IEEE 1451.1 in a Wireless Environment

19



Lessons Learned on Implementation

- Large number of deployed NCAPs could be problematic for system configuration, initialization, and maintenance unless highly sophisticated configuration tools are developed. However, software vendors can provide those tools.
- Modern middleware provides similar service-oriented class characteristics; an OS adaptation layer standard definition approach would ease integrating application functionality.

Implementing IEEE 1451.1 in a Wireless Environment

20

Lessons Learned on Implementation



- The backward or legacy approach to preserving other device bus application interaction complicates and makes the standard over specify itself.
- Implementations of the standard (in C++) are cumbersome and very large for many small embedded systems. A subset standard (IEEE 1451.1 Lite) should be considered for the applications that only need a small subset of the current 1451.1 standard specification.
- NIST implementations unique use of CDR streams as the underlying data typing mechanism provides a less complex asynchronous and synchronous communication marshaling routines.

Implementing IEEE 1451.1 in a Wireless Environment

21

IEEE 1451.1 Benefits



- Using P1451.1 provides:
 - ◆ an extensible object-oriented model for smart transducer application development and deployment
 - ◆ application portability achieved through agreed upon application programming interfaces (API)
 - ◆ network neutral interface allows the same application to be plug-and-play across multiple network technologies
 - ◆ leverages existing networking technology, does not re-implement any control network software or protocols
 - ◆ a common software interface to transducer hardware i/o

Implementing IEEE 1451.1 in a Wireless Environment

22

Part 3: Looking at an IEEE 1451.1 Application

- A minimal IEEE 1451.1 application consist of a few classes:
 - ◆ An NCAP Block (consolidates system and communication housekeeping)
 - ◆ A Transducer Block (provides the software connection to the transducer device)
 - ◆ A Function Block (provides the transducer application algorithm (i.e., obtain and multicast temperature data every second)
 - ◆ Parameters (contains the network accessible variables that hold and update the data)
 - ◆ Ports (network communication objects for publishing and subscribing to information or interacting with other NCAPs using client/server

Implementing IEEE 1451.1 in a Wireless Environment

23

A C++ IEEE 1451.1 Application (NCAP Block)

- Creating a NCAP object starts with defining a TCP server port assignment.
- Create a Tag, this is used to identify the NCAP to others on the network
- Build a Dispatch Address for clients to use to talk to NCAP
- Instantiate the Temperature NCAP
- Register the dispatch address with this NCAP
- Initialize the NCAP state
- Tell NCAP to go "Active" or start running

```

////////////////////////////////////
// Start the client NCAP System initialization process.
////////////////////////////////////

// declare a return code object for 1451 methods
OpReturnCode ret;

// build a local client-server address & port based on this NCAP
ObjectTag localHost;
UInteger16 localPort;
::configLocalAddress(localHost, localPort);

// need an object tag for the client-server NCAP odc
ObjectTag ncapTag("tempNCAP-NCAPBlock");

// create a generic client-server object dispatch address for this NCAP
//
// the port is dummed to 10002 to allow local debugging on the same
// machine of the tempncap <--> configtool
ObjectDispatchAddress ncapaddr(localHost.fast_resp(), 10002/*localPort*/, ncapTag);

// create a set of object properties including the ncap client-server address
ObjectProperties ncapprops(ncapTag, "owner", ncapaddr, "ncapobj");

// create the NIST NCAP
cout << "Creating a NIST Temperature NCAP ..... " << endl;
tempNCAP* ptempNCAP = new tempNCAP( ncapprops,
                                     "NIST", "N2",
                                     "01a", "NIST MfgID# 85073",
                                     "N2", "011",
                                     "VsWorks5.4");

// Register NCAP with itself to make operations on it Network Visible (NCAP is owner)
ret = ptempNCAP->RegisterObject(*ptempNCAP, *ptempNCAP, ncapaddr);
if (ret.majorReturnCode != MJ_COMPLETED)
    cout << "Could not RegisterObject()" << endl;

// initialize the ncap state (start the thread/task)
ret = ptempNCAP->Initialize();

// put the NCAP into the Active State!
ptempNCAP->GoActive();

```

Implementing IEEE 1451.1 in a Wireless Environment

24

A C++ IEEE 1451.1 Application (Transducer Block)



- Creating a Transducer Block starts with defining a Tag for this object
- Optionally, another Dispatch address can be generated for this NCAP
- Create a set of Object Properties that give this object special identity
- Instantiate the Transducer Block
- Register the Transducer Block with the NCAP
- Initialize the TBlock state
- Tell TBlock to go "Active" or start running

```
// get the state of the NCAP, should be INACTIVE!
NCAPBlockState ncapState;
ret = ptempNCAP->GetNCAPBlockState(ncapState);

// print state information
if (ret.majorReturnCode == MJ_COMPLETED)
    cout << "NCAP Block State = " << ((ncapState == 4) ? "NR_INITIALIZED" : "OTHER") << endl;

// tell ncap to go active now
//ptempNCAP->GoActive();

// create a transducer block
ObjectTag tblockTag("tempNCAP-TBlock");
ObjectDispatchAddress tblockAddr(localHost.fast_rep(), 10003, tblockTag);

// create a set of object properties including the ncap multicast address
// first entry of object properties is what registerObject keys on
ObjectProperties tblockProps(tblockTag, "owner", tblockAddr, "tblockobj");

// create a specialized transducer block for the temperature NCAP
cout << "Creating a generic Transducer Block ..." << endl;
tempTBlock *ptempTBlock = new tempTBlock(tblockProps, ptempNCAP);

// Register NCAP with itself to make operations on it Network Visible (NCAP is owner)
ret = ptempNCAP->RegisterObject(*ptempTBlock, *ptempNCAP, tblockAddr);
if (ret.majorReturnCode != MJ_COMPLETED)
    cout << "Could not RegisterObject()" << endl;

// initialize the ncap state (start the thread/task)
ret = ptempTBlock->Initialize();

// prime the tblock application before starting
ptempTBlock->GoActive();

// create a function block
ObjectTag fblockTag("tempNCAP-FBlock");
```

Implementing IEEE 1451.1 in a Wireless Environment

25

A C++ IEEE 1451.1 Application (Function Block)



- Creating a Function Block starts with defining a Tag for this object
- Optionally, another Dispatch address can be generated for this NCAP
- Create a set of Object Properties that give this object special identity
- Instantiate the Function Block
- Register the Function Block with the NCAP
- Initialize the Function Block
- Tell Transducer Block to go "Active"
- Manually "Start()" the application or do it later from a network invocation

```
// create a function block
ObjectTag fblockTag("tempNCAP-FBlock");
ObjectDispatchAddress fblockAddr(localHost.fast_rep(), 10003, fblockTag);

// create a set of object properties including the ncap multicast address
// first entry of object properties is what registerObject keys on
ObjectProperties fblockProps(fblockTag, "owner", fblockAddr, "fblockobj");

// create a generic function block for the application
cout << "Creating a Publisher Function Block ..." << endl;
tempFBlock *ptempFBlock = new tempFBlock(fblockProps, ptempTBlock);

//ptempFBlock->ptempFBlock(fblockProps, ptempTBlock);

// Register the FBlock to make operations on it Network Visible (NCAP block is owner)
ret = ptempNCAP->RegisterObject(*ptempFBlock, *ptempNCAP, fblockAddr);
if (ret.majorReturnCode != MJ_COMPLETED)
    cout << "Could not RegisterObject()" << endl;

// initialize the ncap state (start the thread/task)
ret = ptempFBlock->Initialize();

// prime the fblock application before starting
ptempFBlock->GoActive();

// now actually start the application
ptempFBlock->Start();

// get the state of the NCAP again, should be ACTIVE!
ret = ptempNCAP->GetNCAPBlockState(ncapState);

// print state information
if (ret.majorReturnCode == MJ_COMPLETED)
    cout << "NCAP Block State = " << ((ncapState == 4) ? "NR_INITIALIZED" : "OTHER") << endl;

// process events in the ieee environment
NCAP_WAIT_ON_EVENTS
```

Implementing IEEE 1451.1 in a Wireless Environment

26

A C++ IEEE 1451.1 Application (Ports)

- Ports are used throughout the application in the NCAP, Function Block and Transducer Block
- This Port snapshot is used to multicast temperature data from within the Function Block
- A Publication Topic and Key are used to define this on the network
- Instantiate the Port, in this case an Event Generator Port
- Set timer parameters for how often to "fire" this event

```
void tempFBlock::SetupFBlockData()
{
    // declare a return code object for ieee 1451 method calls
    OpReturnCode ret;

    ObjectTag mtag("multicast tag");

    // create a general multicast object dispatch address for this NCAP
    ObjectDispatchAddress mcast_addr(
        ACE_DEFAULT_MULTICAST_ADDR,
        ACE_DEFAULT_MULTICAST_PORT,
        mtag);

    // create a set of object properties including the ncap multicast address
    ObjectProperties mcast_props("temp fblock", "owner", mcast_addr, "fblockobj");

    // Build PSK_PHYSICAL_PARAMETRIC_DATA
    // =====
    // create a publisher topic for FBlock Data
    topic = new PublicationTopic("TempFBlock-Data");

    // create a event publisher port for FBlock Data
    cout << "Creating a EventGeneratorPublisherPort ..." << endl;
    fblock_data_pub_port = new IEEE1451_EventGeneratorPublisherPort(mcast_props, *topic);

    // set the PubSub key to PSK_PHYSICAL_PARAMETRIC_DATA, meaning data
    fblock_data_pub_port->SetPublicationKey(PSK_PHYSICAL_PARAMETRIC_DATA);

    // event pub has been retrofitted (non-std) to include time-based event generation

    // create a time period for publishing
    TimeRepresentation timerep;
    timerep.seconds = 1;
    timerep.nanoseconds = 0;

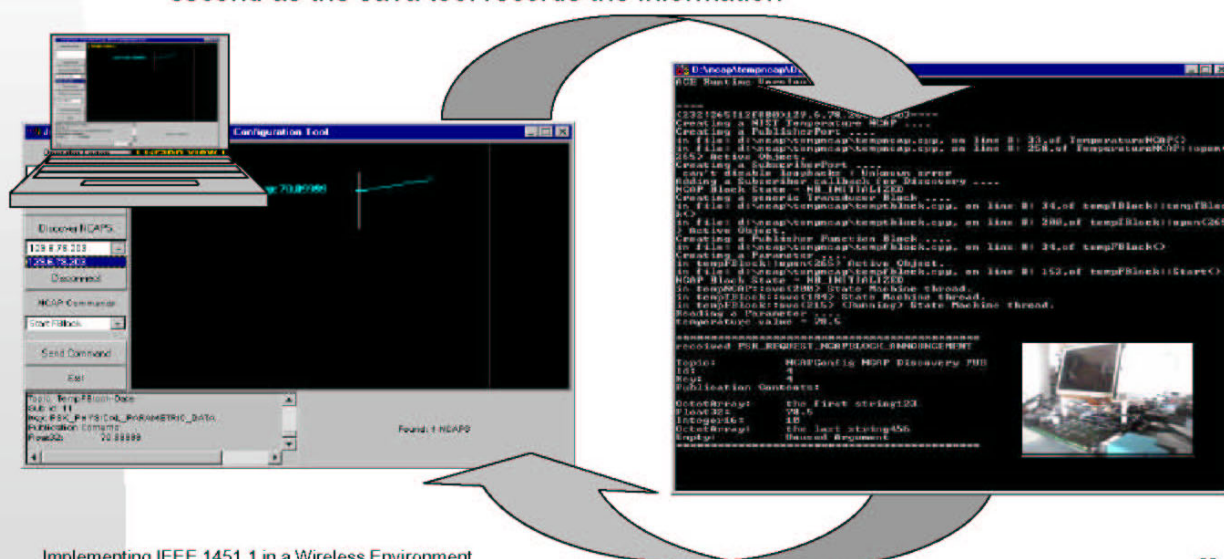
    // set the timer in the eventpublisher
    fblock_data_pub_port->SetTimer(timerep);
}
```

Implementing IEEE 1451.1 in a Wireless Environment

27

Executing an IEEE 1451.1 Application

- An embedded Temperature NCAP Application is running from a remote location on the NIST Intranet
 - As part of the system configuration, a NIST developed Java tool on a Notebook issues a discovery multicast, finds the NCAP, and starts the remote NCAP's Function Block
 - The remote NCAP Function Block responds by publishing temperature data every second as the Java tool records the information



Implementing IEEE 1451.1 in a Wireless Environment

28

Part 4: Using IEEE 1451.1 in a Wireless Environment



- The NIST C++ IEEE 1451.1 reference implementation uses TCP/IP as its underlying control network.
- From TCP/IP, IP multicast and TCP unicast features are used to implement publish/subscribe and client/server, respectively
- ACE is used to abstract the networking code from the application; therefore it is highly adaptive to various protocols
- Wired 802.3 Ethernet has been used primarily for testing. No changes were needed in ACE to support this protocol.
- Wireless 802.11b (11Mbps) Ethernet has also been used for testing. Again, no changes were made to ACE as the TCP/IP protocol is compatible with both 802.3 and 802.11b physical mediums.

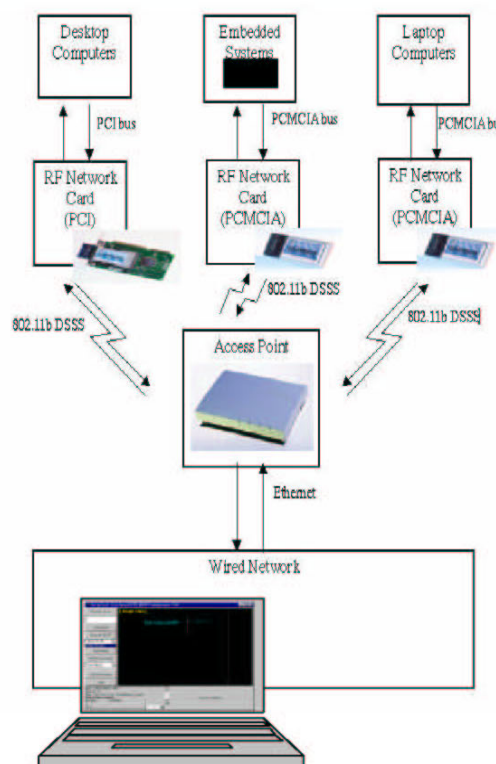
Implementing IEEE 1451.1 in a Wireless Environment

29

Using IEEE 1451.1 in a Wireless Environment



- Testing scenarios included using a wired subnet connected to a wireless extension of the subnet
- Wireless extension uses an Agere (formerly Lucent) Orinoco AP-1000 dual card “access point”
- Range extender antennas are also connected to the access point and each PC-CARD
- Each node on the wireless side executes an IEEE 1451.1 NCAP application
- Java Configuration tool executes beyond the wireless net on the wired subnet



Implementing IEEE 1451.1 in a Wireless Environment

30

Using IEEE 1451.1 in a Wireless Environment



- Limited testing scenarios confined to the office environment at NIST
- Experience has not been good due to range problems within our building.
- Structural barriers within the walls severely restrict the range available from NCAP to access point
- Building constructed in the 60's of solid concrete block with concrete floors, offices
- Range using a wireless subnet is no greater than 20-25' with antenna
- High-end of 11 Mbps range only reached with 15'
- All applications worked as advertised albeit with limited range.
- Multicasting through the bridge needed to be configured through the access point
- Noticeable delay for multicast depending on how configured at access point

Summary



- IEEE 1451.1 is a comprehensive and large standard that adequately addresses the smart transducer industry need for portability and network independent access.
- The standard however in addressing all the facets of the smart transducer is complex and quite large.
- NIST has embarked on implementing a good deal of the standard with emphasis on getting the communication and infrastructure code in place in order to start using the code.
- Choosing and implementing the standard with a solid object-oriented framework such as ACE provides a robust environment for real-time network communication.
- Migrating the implementation to other middleware such as CORBA for heavier weight uses will be reasonable to do
- Several projects at NIST will use the implementation for supporting manufacturing related activities

Implementing IEEE 1451.1 in a Wireless Environment

32

Summary (cont)



- Continued testing in the wireless space is required to gauge the effectiveness of the implementation.
- Bluetooth trials are forthcoming; however, the lack of multicast support will severely impact the applications – continued research here is a must
- Other lightweight middleware packages are going to be isolated – XML and SOAP, etc; however, these protocols do not support asynchronous messaging or publish subscribe in efficient ways
- Slimmer implementations of the IEEE 1451.1 will need to be experimented with for use with the smaller micro platforms.

Implementing IEEE 1451.1 in a Wireless Environment

33

References



- More information about ACE can be found at:
www.cs.wustl.edu/~schmidt/ACE.html
- **IEEE 1451.1-1999 Standard for a Smart Transducer Interface for Sensors and Actuators - Network Capable Application Processor Information Model 2000, ISBN 0-7381-1768-4**

Bluetooth

Small Sensor Area Networks (SANs)

Thurston Brooks

with contributions by
Ericsson Mobile Communications AB

Sensors Expo Chicago 2001

June 4, 2001

T. Brooks 1

Wireless Advantages

- Augment wired LANS
- Minimize setup requirements by installing preconfigured WLANs without MIS support
- Factory floor can exchange data with central databases
- “Instant” reconfigurability
- Installation simplicity and flexibility
- Cost effective TOC
- Mobility-WLANs can provide access to real-time info on-the-go
- Scalability

Sensors Expo Chicago 2001

June 4, 2001

T. Brooks 2

Wireless Advantages (Cont.)

- Eliminates logging and paperwork
- Historical data is easily maintained
- Reduces data gathering errors

Wireless Condition Monitoring

- Will reduce monitoring installation cost
 - \$20/ft typical, as much as \$2000/ft some apps (e.g., Nuclear)
 - Cabling is 30-45% of TOC
- Real-time dynamic range is expensive (correlates to difficulty)
- Spread spectrum technology with data dumps is now feasible for low cost
- Portable
- Immediate alert of alarm conditions and follow-up data dumps are also feasible

Manufacturing Solutions

- Majority of wireless products in marketplace are proprietary spread spectrum and narrow band solutions in the ISM bands (400MHz, 900MHz, 2.4GHz)
 - Today 80% of customers → SS (*Garner Group)
- Typically constrain user to buy from a particular vendor
- Interoperability, low-cost, and broad user base (i.e., market demand) are stimulated by Standards
 - IEEE 802.11 (2.4GHz @1-2Mbps) LAN
 - Bluetooth (2.4GHz @ .75Mbps) PAN
- Potential Interference between 802.11 and Bluetooth

Narrowband Manufacturing Solutions

- Available since early 1980s
- Low throughput
 - term NB → RF BW typically 12.5-25.0 kHz
- Longest Range
- Low cost for large sites
- Little or no vendor interoperability
- Easily jammed
- Site license required for protected bands (450-470 MHz)
- Large form factor

900 MHz ISM Manufacturing Solutions

- Here Today – many 900 MHz LANs in use
- Data rates of 100-450 Kbps are sufficient for many manufacturing applications
- Typically better range than 2.4 GHz systems
- Very crowded band
 - US: cordless phones, vehicle locators, etc.
 - International: GSM cellular and military systems
- Little interoperability
- Lack international acceptance

2.4 GHz ISM Manufacturing Solutions

- Standards based systems
- Multi-vendor support
- High data rates available
 - IEEE 802.11 10 Mbps
 - Bluetooth 1 Mbps
- World-wide acceptance
- Limited range
 - Attenuation
 - Less power
 - Poor coverage – Increased infrastructure costs

IEEE 802.11b

- Defines wireless LAN interoperability
- Two forms:
 - Frequency Hopping Spread Spectrum (FHSS)
 - 2.4 GHz 79 Channels 1 MHz wide (US)
 - Direct Sequence Spread Spectrum (DSSS)
 - 2.4 GHz 3 non-overlapping Channels 22 MHz wide (US)
 - 2.4 GHz 1 Channel 22 MHz wide (Japan)
- Both techniques limited to 1W max radiated power US

802.11 and Bluetooth Interference Potential

- Both standards use frequency hopping
- Bluetooth is predicted to “jam” 802.11 FHSS operation
 - BT will most likely interrupt an 802.11 transmission many times before the 802.11 device hops to the next frequency (Jim Geier-Wireless Nets)
- FCC does not mediate frequency conflicts in the unlicensed bands
 - IEEE has created the 802.15 Coexistence Task Group 2
- Preliminary analysis by 802.15 Group indicates that 802.11 DSSS is very reliable in the presence of BT
- Solutions
 - BT falls off quickly, avoid close proximity with 802.11 devices
 - Reduce 802.11 radio distance

Advantages of Spread Spectrum

- SS can co-exist with other systems that are already using the same frequency bands
- SS has excellent discrimination of signals in multipath environments
- SS is less interference prone than other techniques particularly in manufacturing environments where “noise” can be severe
- SS in the ISM Bands do not need licensing
- SS combined with TDMA has nearly unlimited capacity

Sensors Expo '01/Nov 2001

Nov 4, 2001

T. Brooks 11

Data Rates Manufacturing Solutions

Frequency	Data rate
400 MHz UHF NB	4.8-19.2 Kbps
900 MHz SS	100-400 Kbps
2.4 GHz SS	1-2 Mbps
2.4 GHz SS (802.11b+)	10 Mbps+
5.7 GHz (Future)	20 Mbps++

Guide to Wireless LAN Technologies, Intermec Tech Corp

Sensors Expo '01/Nov 2001

Nov 4, 2001

T. Brooks 12

Ranges of RF Manufacturing Solutions

Frequency	Indoor Range
400 MHz UHF	300-400 ft.
900 MHz SS	220-350 ft.
2.4 GHz SS 100mW US, Asia, Pacific, Latin America	150-200 ft.
2.4 GHz SS 500mW US, Asia, Pacific, Latin America	200-250 ft.
2.4 GHz SS 100mW Europe	125-150 ft.

Guide to Wireless LAN Technologies, Intermec Tech Corp

Sensors Expo C'Binson 2001

June 4, 2001

T. Brooks 13

Factors Effecting Range

- **Transmitter Power**
 - The FCC regulations permit radiated RF power for unlicensed operation of up to 1 watt (+30dBm) when spread spectrum modulation techniques are used. All ISM band wireless modem and wireless LAN products must adhere to these requirements.
- **Receive Sensitivity**
 - Receiver sensitivity quantifies the ability of a receiver to respond to weak signal levels. Typically stated as Maximum Bit Error Rate at a certain RF level, e.g., 10^{-6} BER @ -70dBm
- **Path Loss**
 - Signal attenuation between the transmitter and the receiver.
- **Antenna Gain**
 - Apparent increase in signal strength due to beam forming.

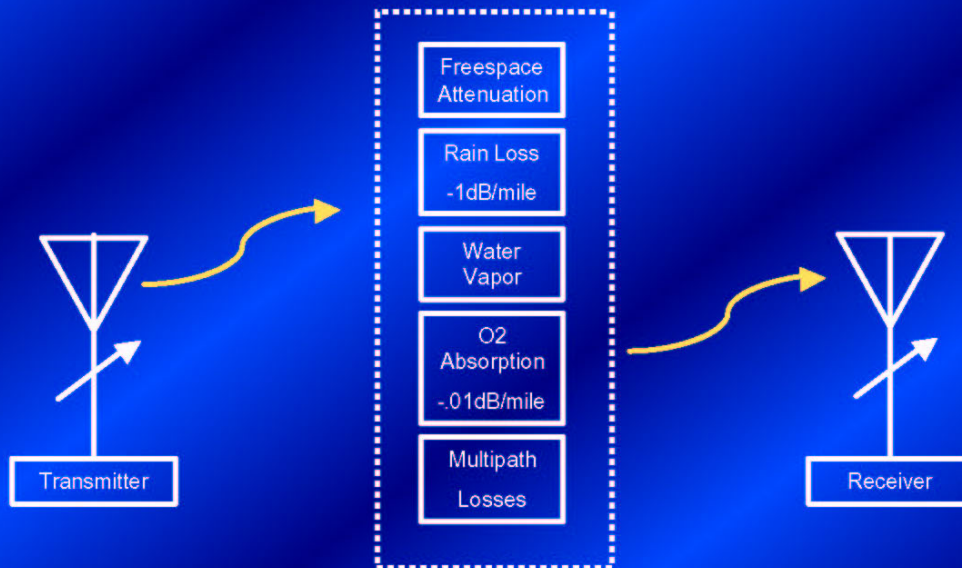
Slide Courtesy of Robert Haller, Grayhill

Sensors Expo C'Binson 2001

June 4, 2001

T. Brooks 14

Wireless Attenuation

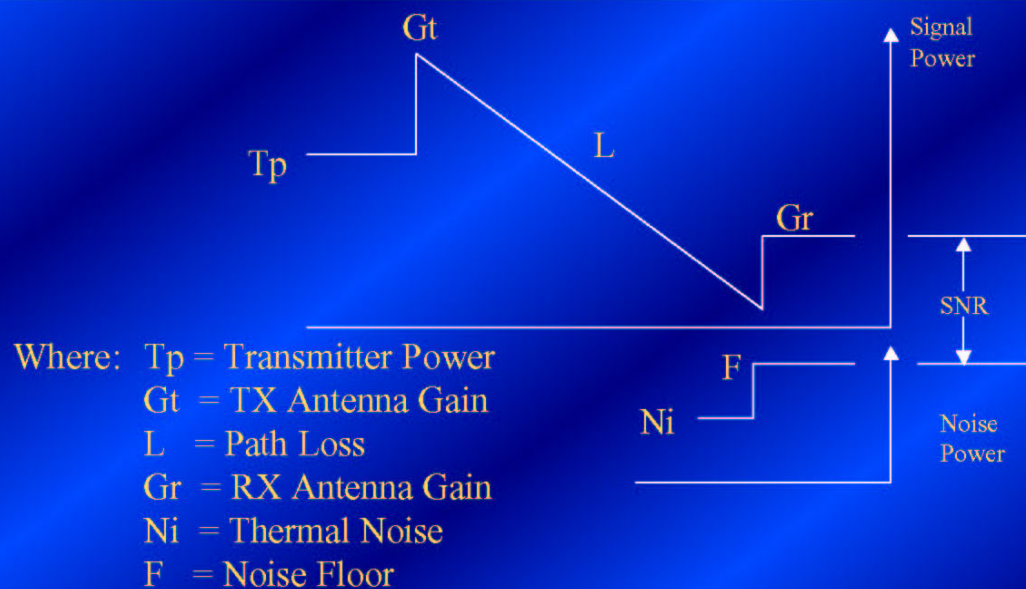


Sensors Expo Chicago 2001

June 4, 2001

T. Brooks 15

Range - Link Budget



Slide Courtesy of Robert Haller, Grayhill

Sensors Expo Chicago 2001

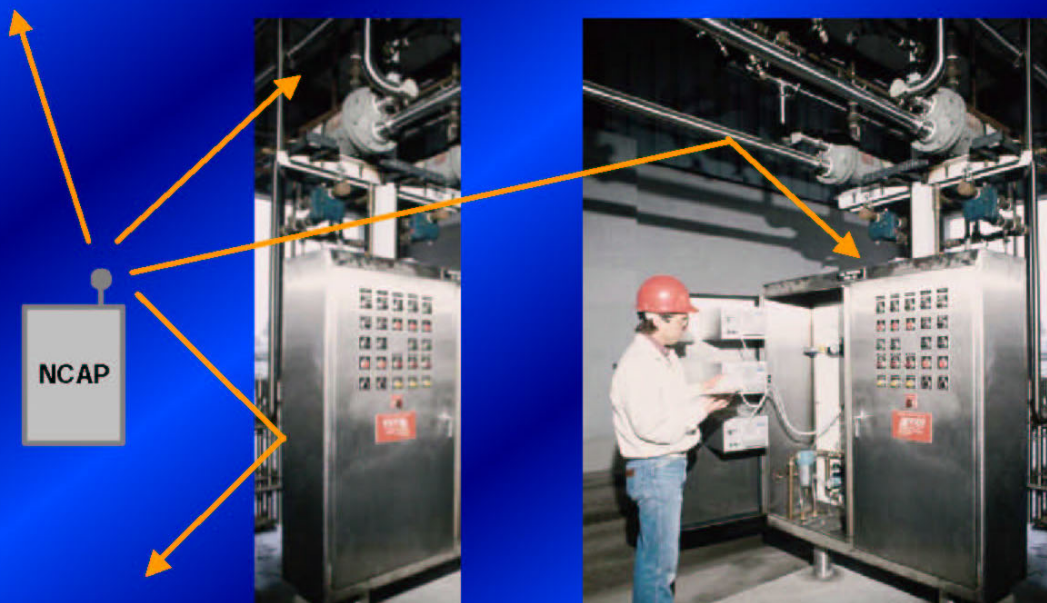
June 4, 2001

T. Brooks 16

Antenna Fundamentals

- **Omnidirectional (Isotropic Source)**
 - An isotropic source is a source radiating energy equally in all directions.
- **Directivity**
 - Directivity is the ratio between the maximum radiation intensity from a transmitter and the radiation intensity from an isotropic source radiating the same power.
- **Gain**
 - The gain of an antenna is a function of the antenna's directivity – as the antenna's beam is focused it's Effective Radiated Power (ERP) is increased.

Multipath



Ray Tracing

- Software modeling of 2D or 3D ray tracing is the most reliable.
- System using only free-space path loss are the least reliable
- Modeling has limited value within a manufacturing plant

SensorsExpo Chicago 2001

June 4, 2001

T. Brooks 19

Applications

- Retail
 - Wireless link to database
 - Transaction based
 - Inventory control
- Warehouses
 - Handheld devices improve efficiency
 - Eliminate paper
 - USPS logs trucks in and out of distribution centers
- Healthcare
 - Patient monitoring
 - Tracking pharmaceuticals

SensorsExpo Chicago 2001

June 4, 2001

T. Brooks 20

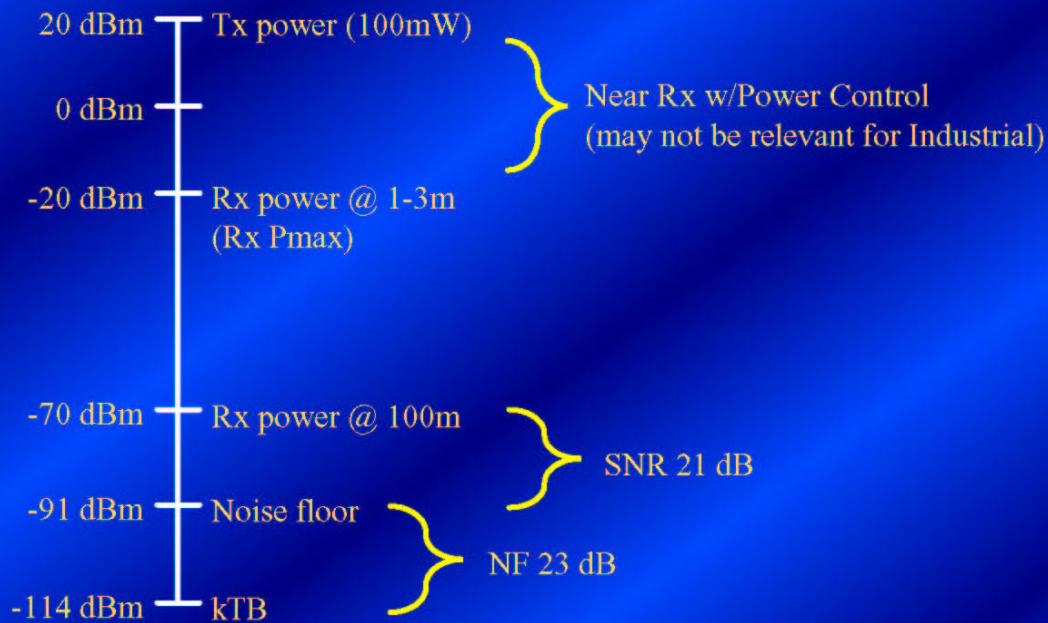
Applications (Cont.)

- Restaurants
- Utilities
 - Meter reading
 - Handheld devices improve efficiency
 - Eliminates paper
 - Kansas City Power & Light system monitors 150,000 customers, automatically tracks usage and issues bills – no meter readers
 - Monitor electrical distribution
- Vending
 - Typically utilize existing cellular networks

Applications (Cont.)

- Traffic Monitoring
- Industrial Process Plants
 - Tank Farms
 - Monitor fluid levels, flow rates, temperatures
 - LOS 2 miles
- Ship/Transportation Crane Controls (Omnex)
- Manufacturing Cells

Bluetooth Link Budget for Max Power TX



Sensors Expo Chicago 2001

June 4, 2001

T. Brooks 23

Operational States

- Standby
- Inquiry-Paging
- Connections
 - Active
 - Power Conserving
 - Hold (delta t)
 - Sniff (Poling)
 - Park (Listen but do not disturb master)
- Synchronous Connection-Oriented (SCO) Link (primarily used for voice)
- Asynchronous Connection-Less (ACL) Link (primarily used for packet data)

Sensors Expo Chicago 2001

June 4, 2001

T. Brooks 24

Features

- Regulations
 - FCC Part 15 compliance
 - ETSI 300 328
- Frequencies
 - covers all of the 2.4GHz ISM band
- Time Division Duplex
- Stand-by Modes
 - Park
 - Hold
 - Sniff
 - Standby

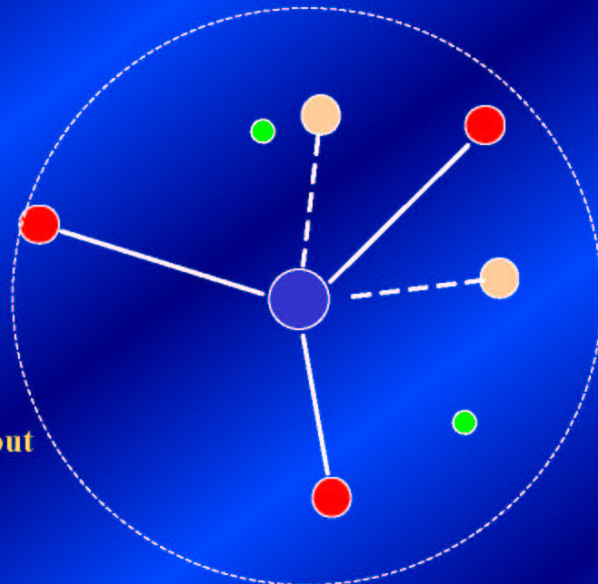
SensorsExpo Chicago 2001

June 4, 2001

T. Brooks 25

Operational States

- **Master – sets the hopping sequence**
- **Active Slave – 3-bit MAC address for immediate communication → only 7 slaves in a piconet**
- **Parked Slave – synchronized but has no MAC address**
- **Standby – Unconnected**

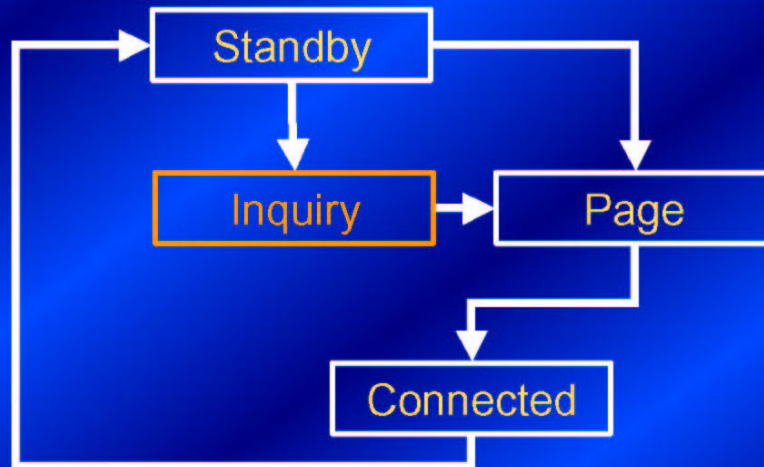


SensorsExpo Chicago 2001

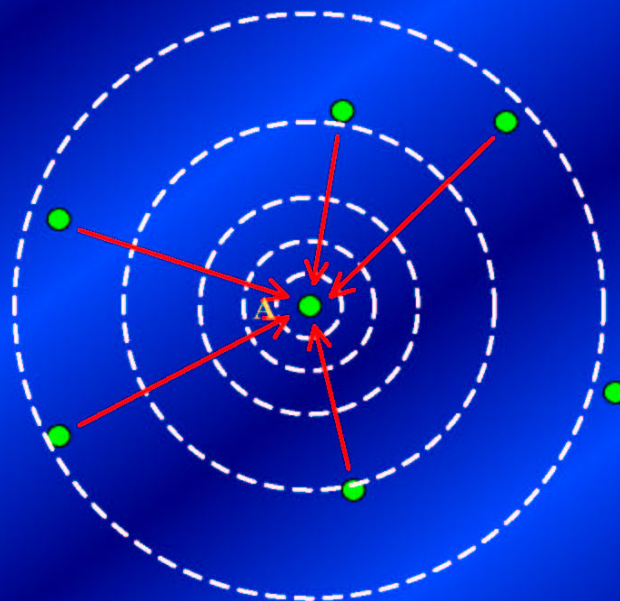
June 4, 2001

T. Brooks 26

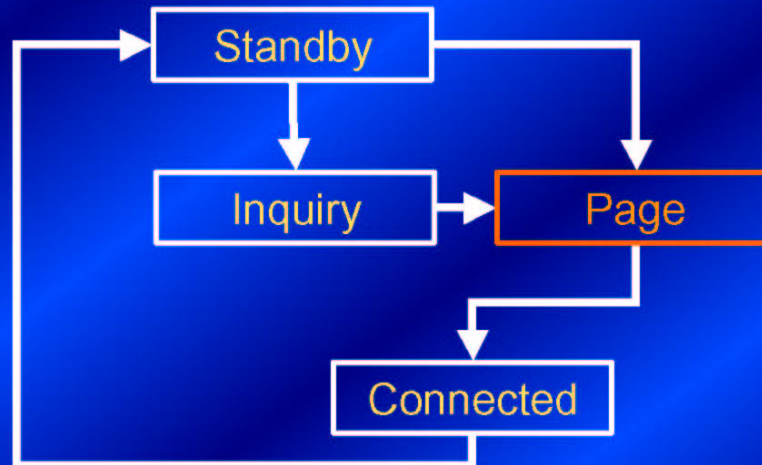
Making a Connection



Inquiry



Making a Connection



SensorsExpo Chicago 2001

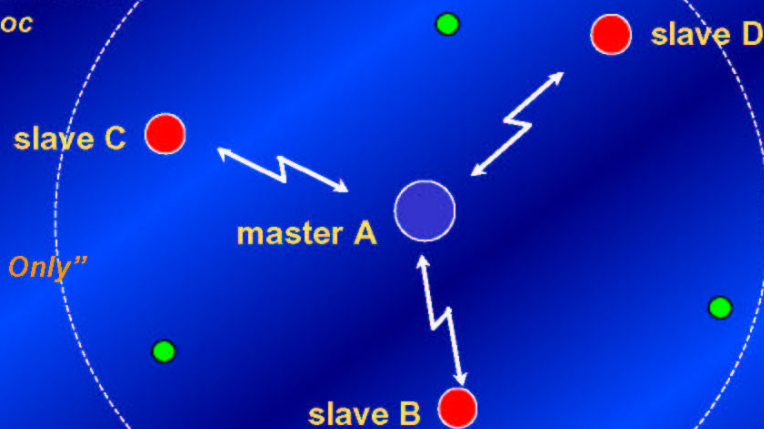
June 4, 2001

T. Brooks 29

Creating a Piconet

Piconet: A collection of devices connected in an ad hoc fashion

"By Invite Only"

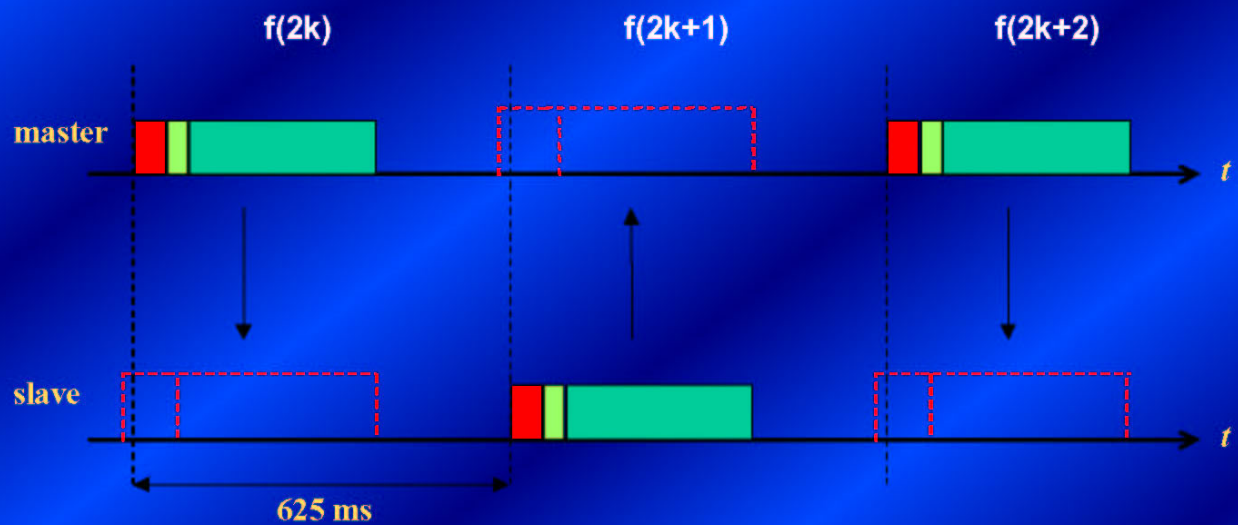


SensorsExpo Chicago 2001

June 4, 2001

T. Brooks 30

FH with TDD Channel

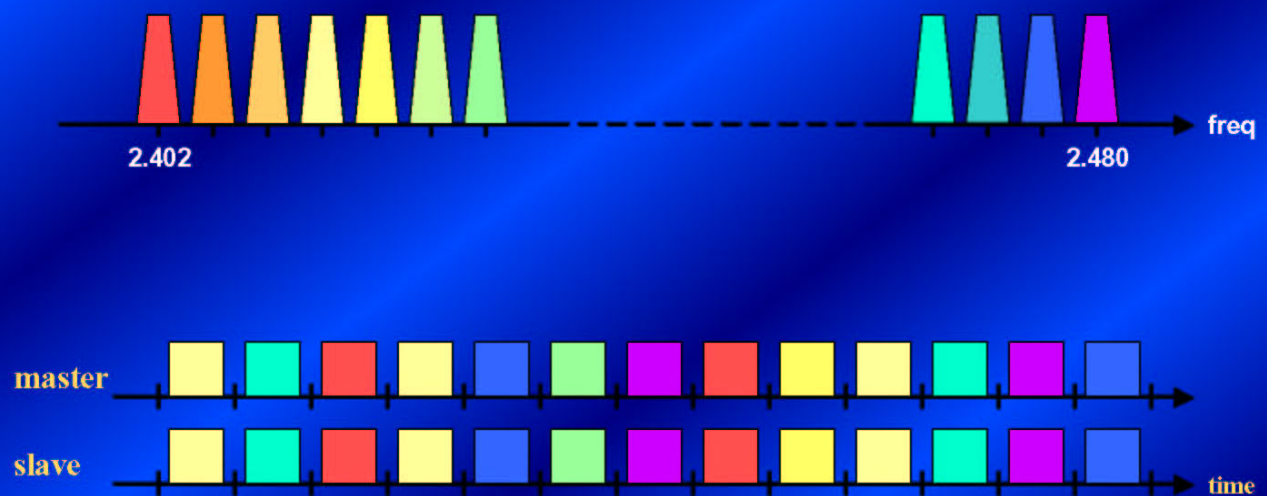


Bluetooth SIG, Inc. 2000

June 4, 2001

T. Brooks, Inc.

Frequency Hopping



Sensors Expo Chicago 2001

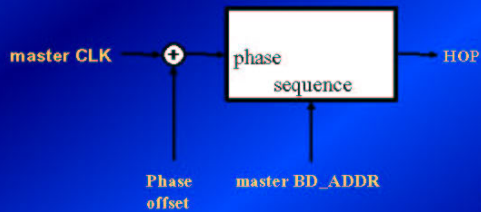
June 4, 2001

T. Brooks 32

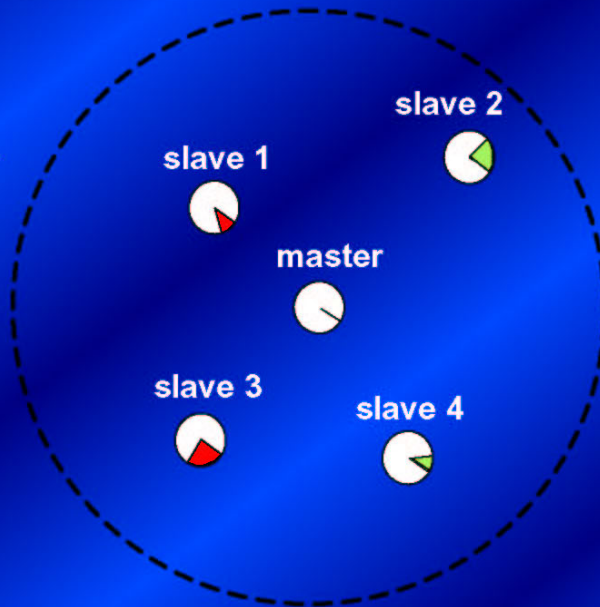
Synchronisation of Physical Channel

master BD_ADDR → hop sequence

master CLOCK → phase



Master Unit is the device in a piconet whose clock and hopping sequence are used to synchronize all other devices in the piconet

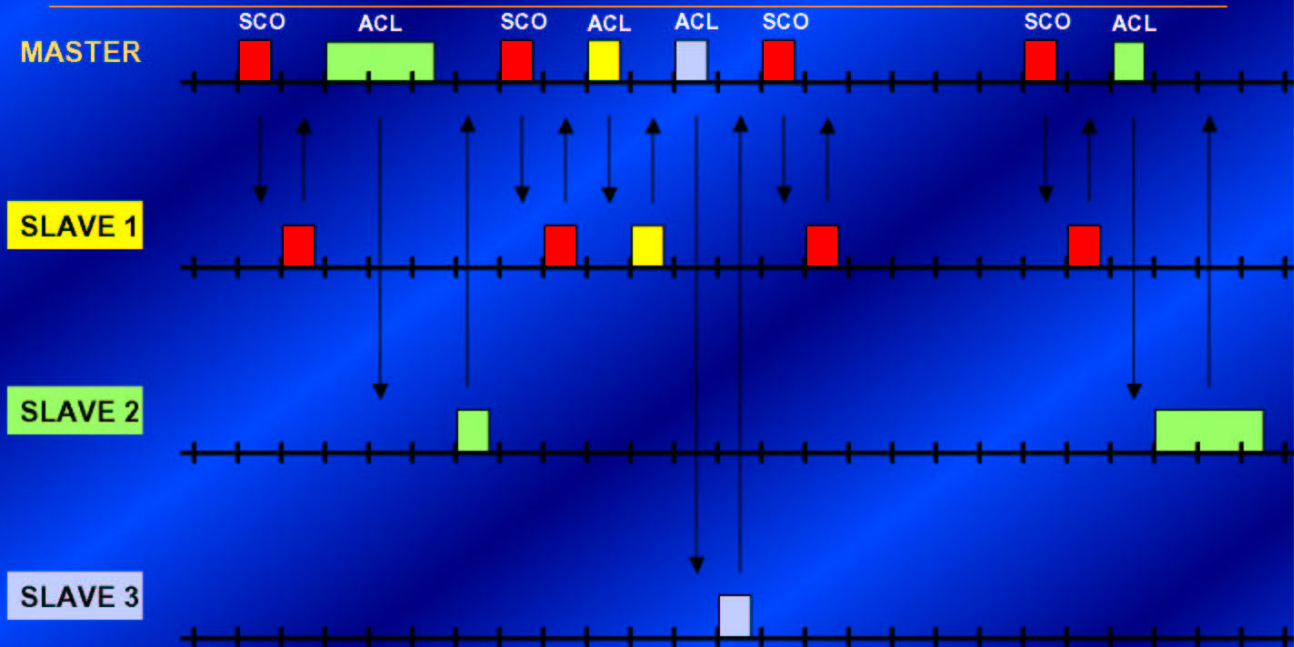


Sensors Expo Chicago 2001

June 4, 2001

T. Brooks 33

Mixed Link Example



Sensors Expo Chicago 2001

June 4, 2001

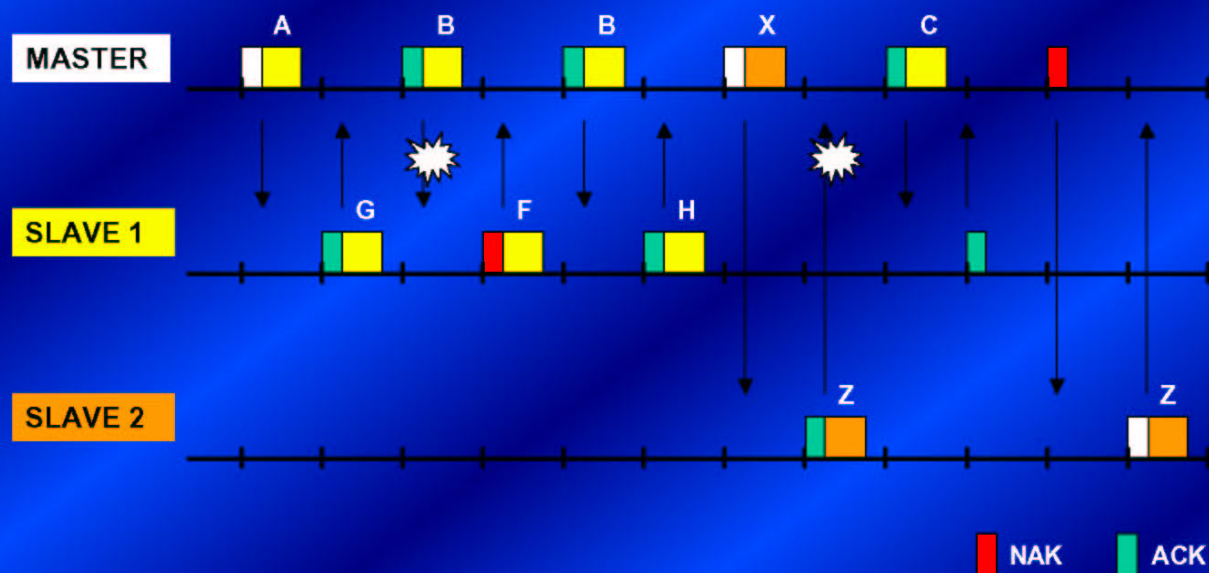
T. Brooks 34

Data Rates (kb/s)

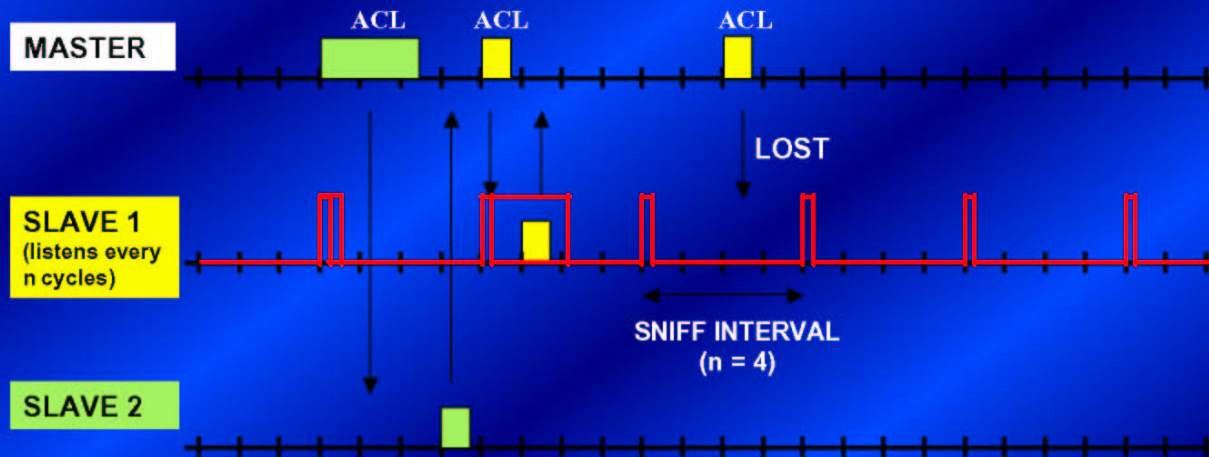
type	symmetric	asymmetric	
DM1	108.8	108.8	108.8
DH1	172.8	172.8	172.8
DM3	258.1	387.2	54.4
DH3	390.4	585.6	86.4
DM5	286.7	477.8	36.3
DH5	433.9	723.2	57.6

Master unit controls the link bandwidth, decides how much is given to each slave, and sets the symmetry of the link

Automatic Repeat Request (ARQ) Scheme



Sniff Mode

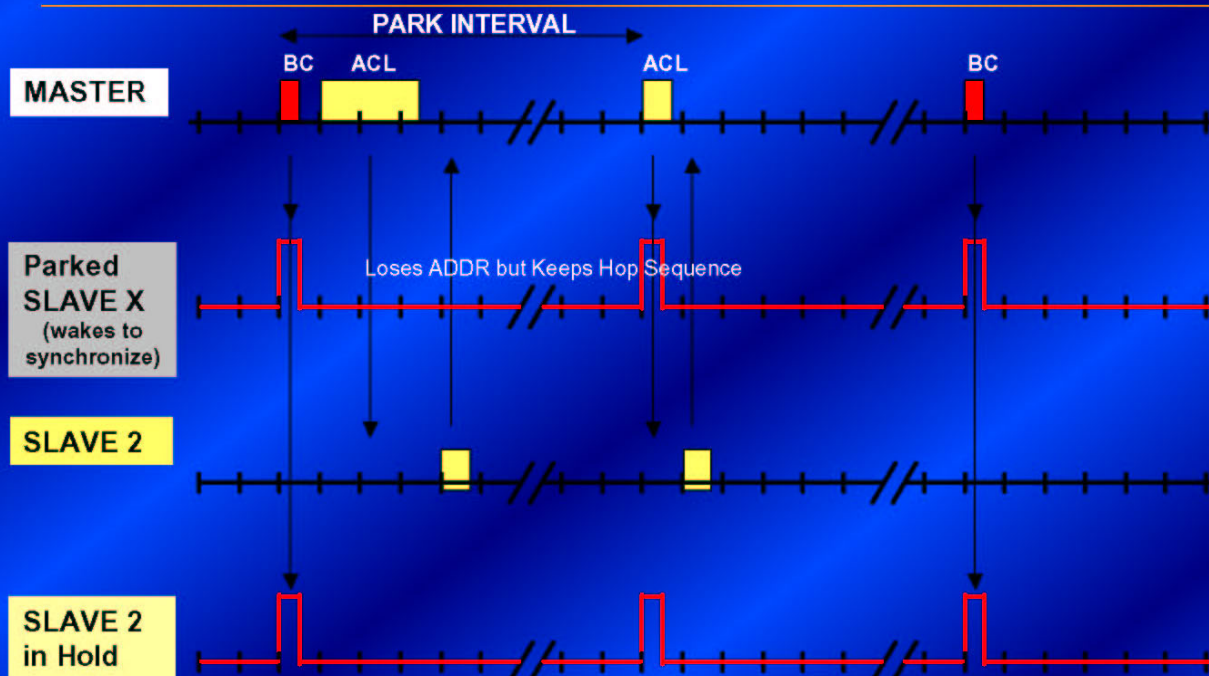


SensorsExpo Chicago 2001

June 4, 2001

T. Brooks 37

Park & Hold Modes

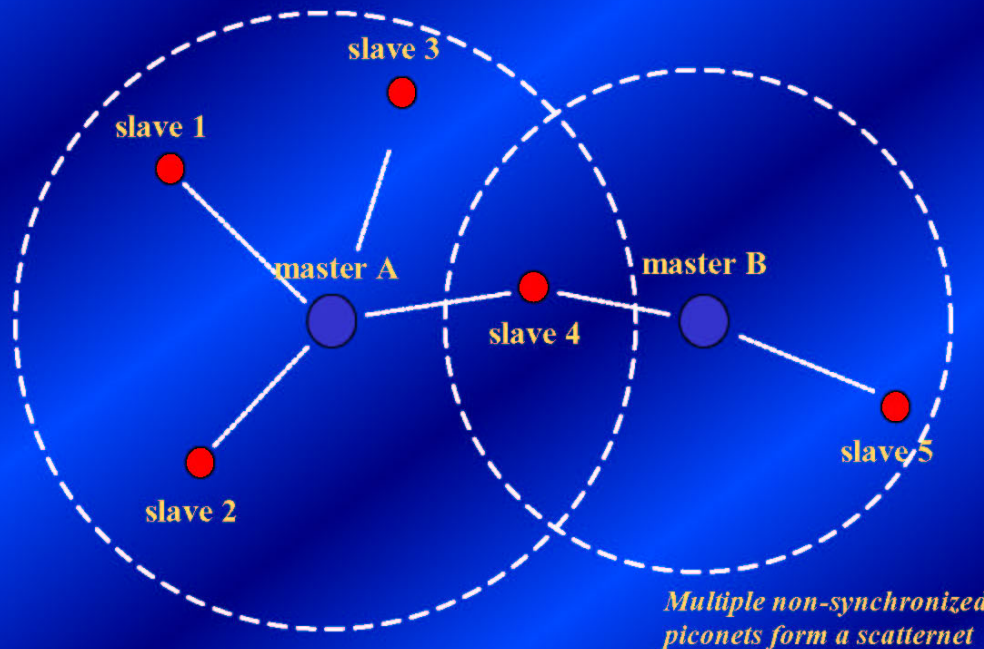


SensorsExpo Chicago 2001

June 4, 2001

T. Brooks 38

Scatternet



Sensors Expo Chicago 2001

June 4, 2001

T. Brooks 39

Bluetooth Today (242 products)



Sensors Expo Chicago 2001

June 4, 2001

T. Brooks 40

Information

- [http:// www.bluetooth.com](http://www.bluetooth.com)
- <http://news.zdnet.co.uk/0,,t294,00.html>
- Any Search on “bluetooth”

Wireless Ethernet (802.11) Overview

James D. Gilsinn

National Institute of Standards & Technology

Intelligent Systems Division

james.gilsinn@nist.gov

Tuesday, July 17, 2001

National Institute of Standards & Technology
Intelligent Systems Division

Slide 1

What is IEEE 802.11?

- IEEE 802.11 is extension of Ethernet standard (IEEE 802.3) into wireless communications
- Allows roaming computers to talk to other devices (peer-to-peer) or connect to wired network (transmitter/receiver)
- IEEE standard allows interoperability between multiple vendors products

Tuesday, July 17, 2001

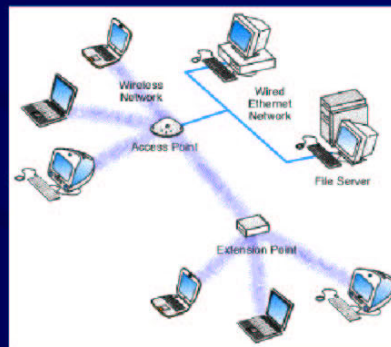
National Institute of Standards & Technology
Intelligent Systems Division

Slide 2

Examples of 802.11 Networks



Peer-to-Peer Network



Transmitter/Receiver
(Wired/Wireless Network)

Pictures from Vicomsoft Web Site, <http://www.vicomsoft.com/>

Tuesday, July 17, 2001

**National Institute of Standards & Technology
Intelligent Systems Division**

Slide 3

802.11 Specification

- Speeds of 1-2 Mb/sec
- Operating Range: 10-100m indoors, 300m outdoors
- Power Output Limited to 1 Watt in U.S.
- Frequency Hopping (FHSS), Direct Sequence (DSSS), & Infrared (IrDA)
 - Networks are NOT compatible with each other
- Uses unlicensed 2.4 GHz band (2.402-2.480 GHz)
- Provide wireless Ethernet for wired networks

Tuesday, July 17, 2001

**National Institute of Standards & Technology
Intelligent Systems Division**

Slide 4

802.11 Variations

- 802.11a
 - Speeds of 6-54 Mb/sec
 - Uses 5 GHz band instead
 - Not commercially available, at the moment
- 802.11b
 - Speeds of 5.5 or 11 Mb/sec
 - Direct Sequence Spread Spectrum (DSSS) Only
 - Backward compatible with 802.11 DSSS components

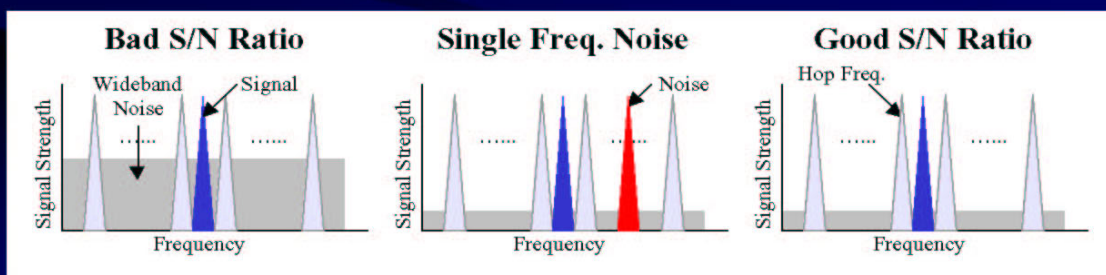
Tuesday, July 17, 2001

National Institute of Standards & Technology
Intelligent Systems Division

Slide 5

Freq. Hopping Spread Spectrum

- Uses 79 separate 1 MHz channels from 2.402-2.480 GHz
- Hops about every 0.1 sec (22 hop pattern, 2.5 hop/sec minimum in US)
- Immune to single frequency noise, has trouble with wideband noise
- Many networks can be located in the same area
- Uses less power to transmit & less expensive to build than DSSS

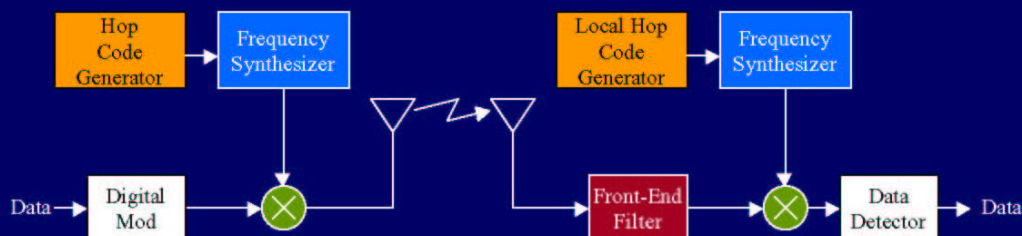


Tuesday, July 17, 2001

National Institute of Standards & Technology
Intelligent Systems Division

Slide 6

FHSS Block Diagram



From Wayne Manges Presentation @ ISA Conference on Wireless Communications

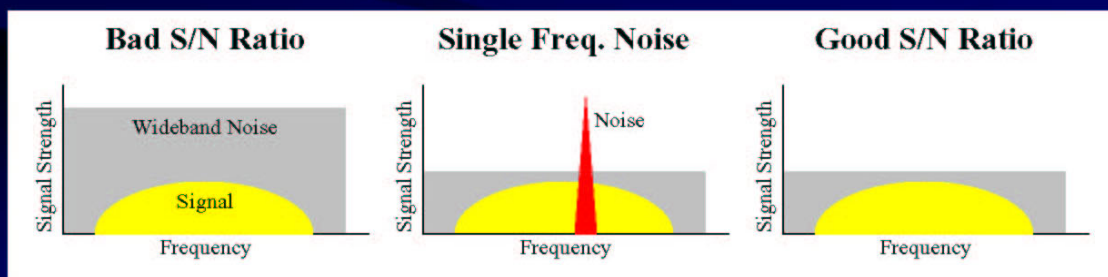
Tuesday, July 17, 2001

**National Institute of Standards & Technology
Intelligent Systems Division**

Slide 7

Direct-Sequence Spread Spectrum

- Signal modulated with a spreading code (11-bit Barker Sequence)
- All 802.11b compliant products use the same spreading code
- Higher data rates because of “fatter pipe” (about 11 MHz)
- Allows for some single frequency noise & higher wideband noise
- Only allows for 3 networks in same area
- Uses higher power to transmit & more expensive to build than FHSS



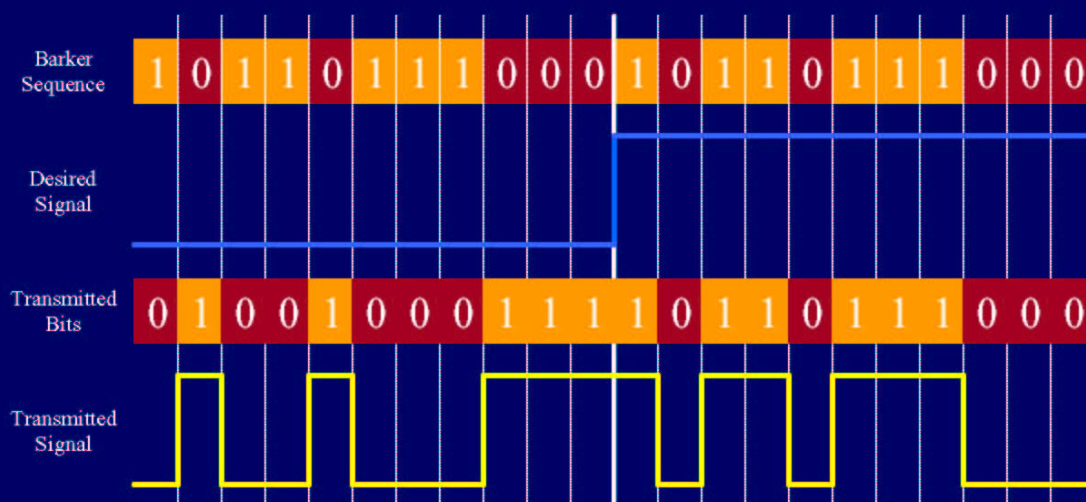
Tuesday, July 17, 2001

**National Institute of Standards & Technology
Intelligent Systems Division**

Slide 8

DSSS (cont'd)

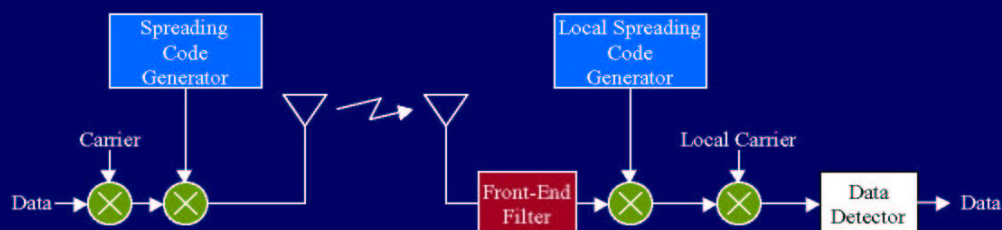
Transmit data bits 01 using DSSS spreading code 10110111000



Tuesday, July 17, 2001
National Institute of Standards & Technology
Intelligent Systems Division

Slide 9

DSSS Block Diagram



From Wayne Manges Presentation @ ISA Conference on Wireless Communications

Tuesday, July 17, 2001
National Institute of Standards & Technology
Intelligent Systems Division

Slide 10

802.11b Security (WEP Protocol)

- 802.11b uses Wired Equivalent Privacy (WEP) protocol for encryption and authentication
- WEP protocol is self-synchronizing
- 64-bit key (40-bit secret code, 24-bit “init” vector)
 - 128-bit keys available for extra \$\$\$
- Additional info sent with message for data integrity checks (i.e. CRC-32)
- Uses same key to encrypt/decrypt message

Tuesday, July 17, 2001

National Institute of Standards & Technology
Intelligent Systems Division

Slide 11

802.11b Equipment

- IEEE standard allows for many vendors
 - Access Point Base Stations \$200-\$1200
 - Access Point Software \$175
 - PCMCIA Cards \$80-\$375
 - PCI Cards \$50-\$300
 - Extension Point Base Stations \$1400
- Prices vary greatly due to different features like security options, number of users, etc.
- <http://www.wirelesscentral.net/>

Tuesday, July 17, 2001

National Institute of Standards & Technology
Intelligent Systems Division

Slide 12

References

- <http://www.cwt.vt.edu/>
- <http://www.unc.edu/depts/oit/ns/wireless/>
- <http://www.vicomsoft.com/>
- <http://www.wirelessethernet.org/>
- <http://www.wlana.com/>
- <http://www.wirelesscentral.net/>
- <http://www.nortelnetworks.com/>
- <http://www.proxim.com/>
- <http://www.cisco.com/>

Wireless Interface Options for 1451

Presented at the
Wireless Sensing Workshop
Sensors Expo /2001
June 4, 2001

Michael R. Moore
Oak Ridge National Laboratory

Intelligent Integrated Sensors & Systems

ornl

Key Elements of Sensor Standard

- TEDS
- Synchronization of Data Sampling
- Unique Identification
- Accessibility to Networks (e.g. Internet)
- Plug-**and**-Play Sensor Connections

Intelligent Integrated Sensors & Systems

ornl

Business And Technology Issues Drive Selection

- Business/Marketing
 - Industry Acceptance
 - » Cost
 - » Availability
 - » Reputation
 - » Form and Function
 - Standards Development
 - » Leverage Current Standards
 - » Buy-in From Membership
 - Future Growth
- Technology
 - Bandwidth
 - Power requirements
 - Compatibility
 - » System Integration
 - » RF Interference
 - # of Nodes
 - Size/Length of Network
 - Extensible for Future Growth

Intelligent Integrated Sensors & Systems

ornl

Candidate 1451 MAC/PHY From Other Wireless Standards (Business Issues)

- Use IEEE 802 Family As a Model
 - Get Market Acceptance of Protocol
 - Let MAC/PHY Adapt to the Market Place
- Time to Market
 - Which MAC/PHY Are Supported by Current ASICs
 - Which Existing Standardized PHY is the Closest to Meet Our Needs
- Cost

Intelligent Integrated Sensors & Systems

ornl

Candidate 1451 MAC/PHY From Other Wireless Standards (Technology Issues)

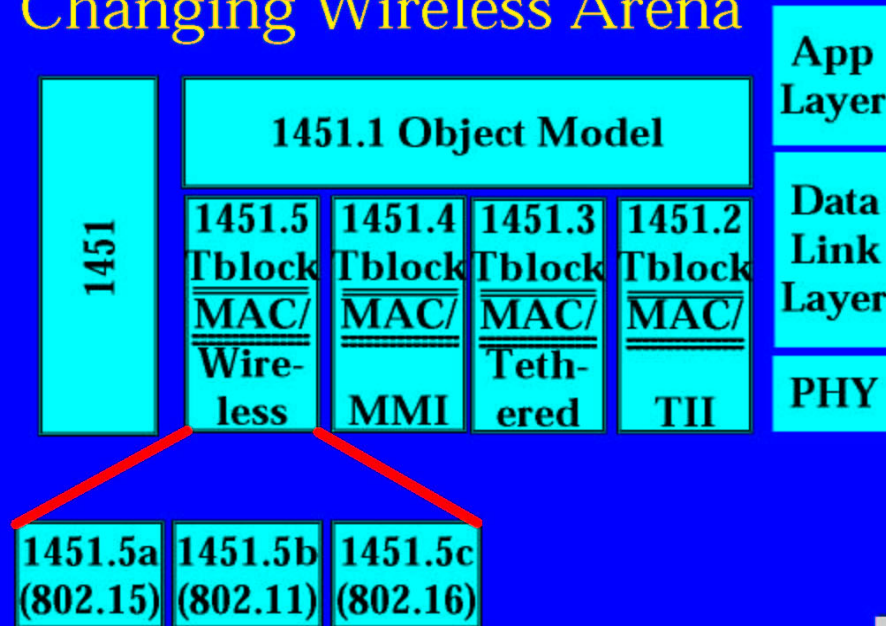
Std	OFDM	FHSS	DSSS	GHz	Size	Mbps
IS-95			x	1 +/-	Cell	0.x
Bluetooth		x		2.45	PAN	1
P802.15		x		2.45	PAN	1
P802.16b	x			5	WAN	54
802.11a	x			5	LAN	54
802.11		x	x	2.45	LAN	1, 2
802.11b			x	2.45	LAN	5.5, 11

IEEE 802 Wireless Projects

- IEEE 802.15 (Bluetooth)
- IEEE 802.11
 - » Clause 14 - 1,2 Mbps FHSS LAN MAC for 2.4 GHz
 - » Clause 15 - 1,2 Mbps DSSS LAN for 2.4 GHz
- IEEE 802.11a
 - » Adds Clause 17 - ≤54 Mbps OFDM LAN for 5 GHz
- IEEE 802.11b
 - » Adds Clause 18 - 5.5 and 11 Mbps DSSS for 2.4 GHz
- IEEE 802.16b (task group 4) Wireless High-Speed Unlicensed Metropolitan Area Network (Wireless HUMAN)
 - » MAC: IEEE 802.16
 - » PHY: IEEE 802.11a; ETSI BRAN HIPERLAN/2

IEEE P1451.5

Could Accommodate Rapidly Changing Wireless Arena



Intelligent Integrated Sensors & Systems

ornl

1451.5 Wireless

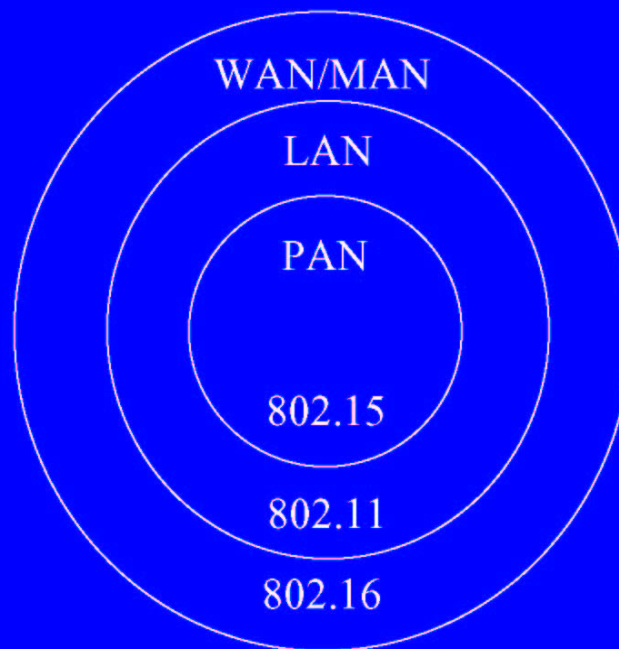
Leverages Other 1451 Projects

- Smart Transducer Object Model from 1451.1
- TEDS Concept from 1451.2
- Synch and XML TEDS from P1451.3
- Compact TEDS and Transducer Interface from P1451.4

Intelligent Integrated Sensors & Systems

ornl

Size of Network vs PHY



Intelligent Integrated Sensors & Systems

ornl

Future Growth

- A Well-Planned Architecture Enables the Quick Incorporation of...
 - Emerging Wireless Technologies
 - Security (as described in IEEE 1363)
 - Mobile Ad-Hoc Networking
 - Others

Intelligent Integrated Sensors & Systems

ornl

Conclusion

- A new Wireless Interface, IEEE P1451.5, is Proposed that will
 - Leverage Sensor Networking Capabilities From the Other IEEE 1451 Projects
 - Include Multiple MAC/PHY Combinations
 - Use IEEE 802 Approach as a Model
 - Has a Layered Architecture
 - » Enables Rapid Ramp-Up Utilizing Existing Products
 - » Enables Future Growth

Developing Executable Specifications for Networking Smart Transducers to Bluetooth

Steven B. Bibyk
Information Electronics
The Ohio State University
bibyk.1@osu.edu

15 May 2001

Department of Electrical Engineering



Problem
Statement

Problem Statement

- Networking smart transducers wirelessly is attractive
- For small-span networks, using existing wireless technology is simpler, quicker and inexpensive
- Challenge is to design a *network infrastructure* according to voluminous, complicated standards and new commercial technology



Wireless Sensor Standards

- Combining Sensor Standards with Wireless Standards is error prone. Key requirements are nebulous --- battery life?
- Need to be able to exercise & forecast how the specifications will play out:
 - System Level Definition Language
www.sldl.org (get white paper)
- Hardware Description Language (VHDL)



Department of Electrical Engineering



NetEnabled Overview

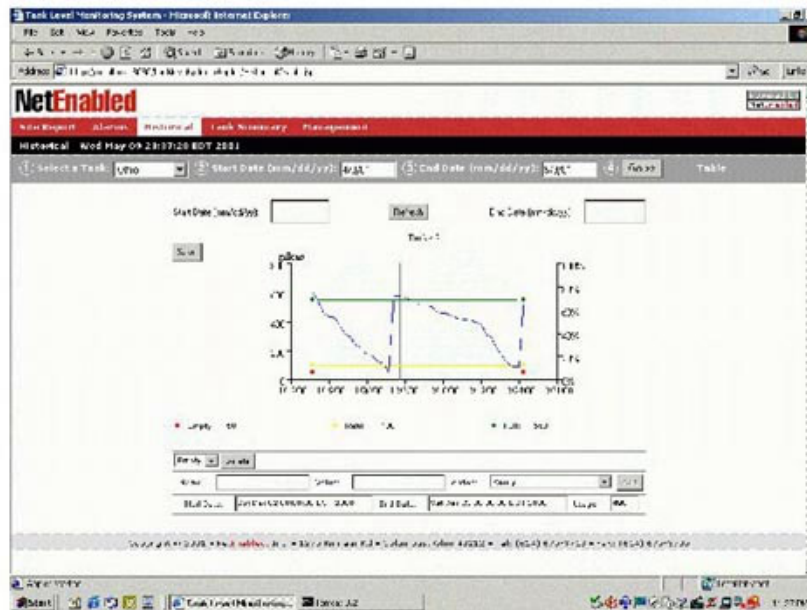
- Connecting Devices to:
 - Networks
 - Data centers
 - Desktops
- Using:
 - Serial, Ethernet, RF, TCP/IP, cellular, CDPD, satellite, pager
- Need:
 - RF technologies supporting:
 - 4+ years for hourly readings using only 3.6V, 5000mAh battery
 - Variable transmissions power and distance
 - Fast sync times and short protocol negotiations
 - Application: Class I, Div I environments



Department of Electrical Engineering



MyTankLevels.com



Department of Electrical Engineering

ASIC+

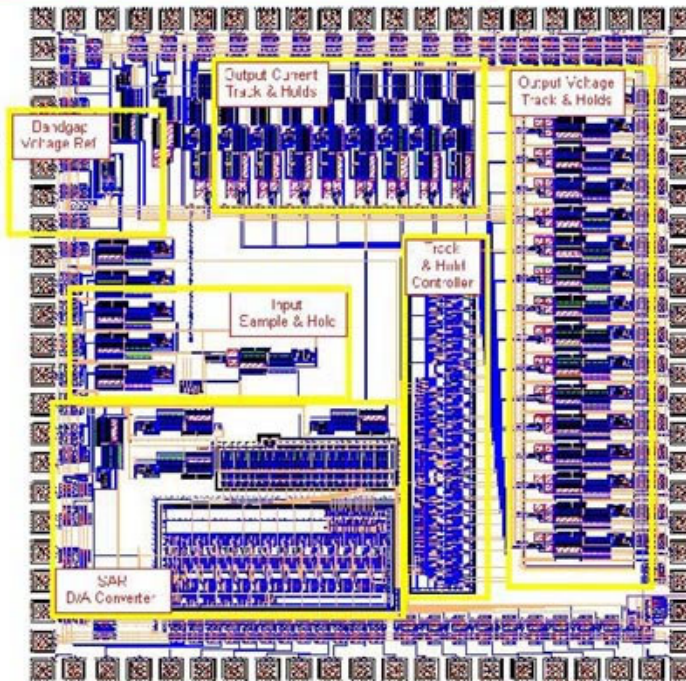
- Electronics Design Business
- Specialty in Mixed Signal ASIC tapeouts.
- RF CMOS chip design and delivery.
- Embedded Systems and Firmware.
- Strong Methodology in using Hardware Description Languages - mainly for digital systems but also for Analog
- IEEE 1076 (1987, '93, '99) VHDL-AMS

Department of Electrical Engineering

82

Past Designs and Customers

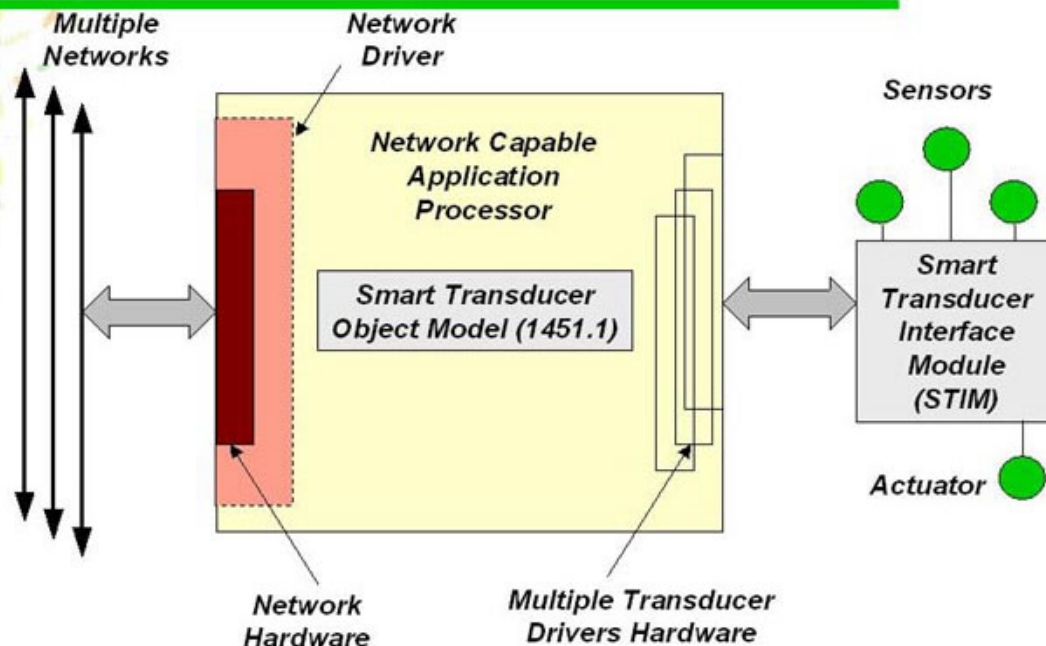
- System (SoC)
- RF Sensor Codec
- Satellite Control
- RF ID Tag
- UWB
- contact:
- Todd James



Department of Electrical Engineering

Background

IEEE 1451.1 Standard

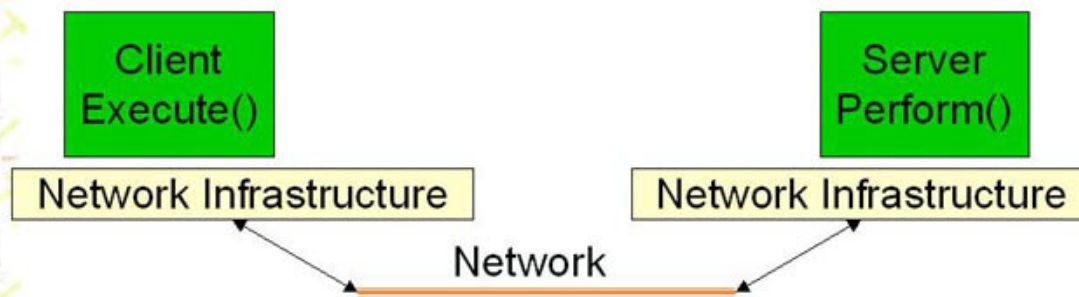


Department of Electrical Engineering

IEEE 1451.1 Client-Server Network Model

Background

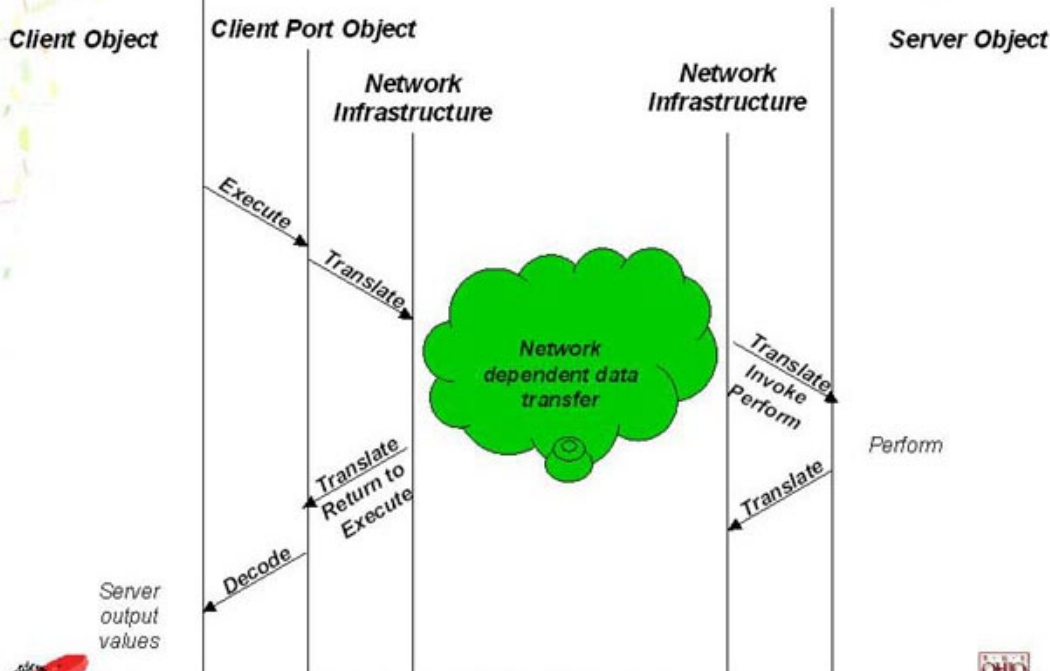
- Tightly coupled, point-to-point model
- NetInf translates IEEE 1451 datums to network specific format



Department of Electrical Engineering



IEEE 1451 Client-Server Model

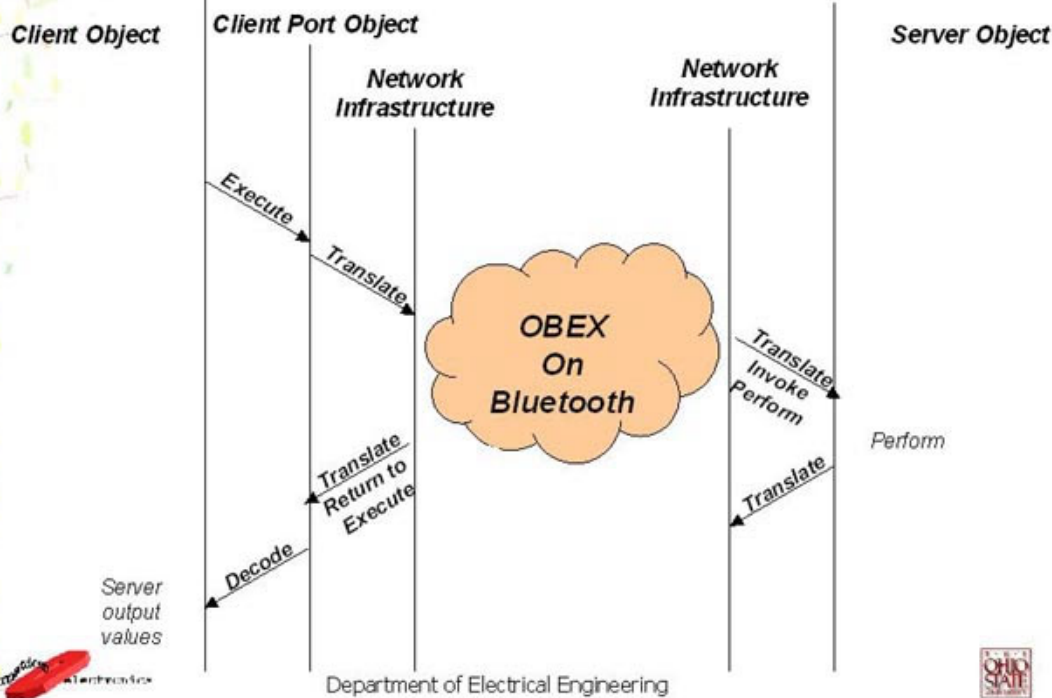


Department of Electrical Engineering



IEEE 1451 Communication on Bluetooth

Problem
Statement



VHDL Design of Interface Entity

Problem
Statement

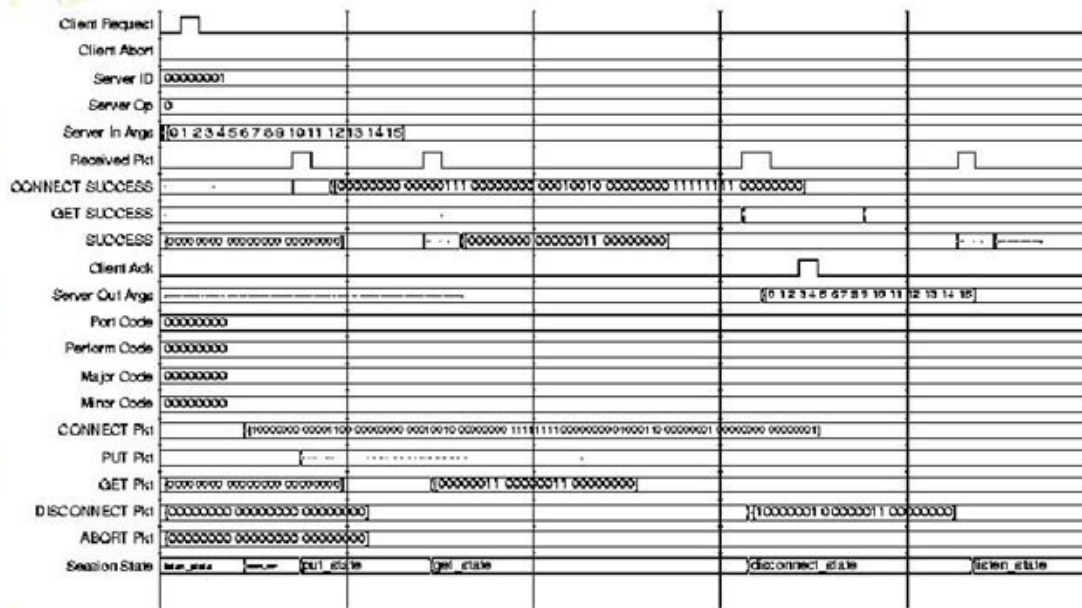
```

ENTITY interface IS
PORT(
    client_request : IN BIT;
    client_abort : IN BIT;
    serverid : IN UInteger8;
    server_operation : IN INTEGER;
    server_input_args : IN integer_array;
    received_pkt : IN BIT;
    connect_success_pkt : IN byte_array(0 TO 6);
    get_success_pkt : IN byte_array(0 TO payload_packet_length);
    success_pkt : IN byte_array(0 TO 2);
    client_ack : OUT BIT;
    server_output_args : OUT integer_array;
    portcode : OUT UInteger8;
    performcode : OUT UInteger8;
    majorcode : OUT UInteger8;
    minorcode : OUT UInteger8;
    connect_pkt : OUT byte_array(0 TO 10);
    put_pkt : OUT byte_array(0 TO payload_packet_length);
    get_pkt : OUT byte_array(0 TO 2);
    disconnect_pkt : OUT byte_array(0 TO 2);
    abort_pkt : OUT byte_array(0 TO 2);
    session_state : OUT state);
END interface;
    
```

Department of Electrical Engineering

Example VHDL Simulation

Problem
Statement



Department of Electrical Engineering

Methodology

- Design the requirements of a network infrastructure.
 - Detailed study of IEEE 1451 standard
 - Evaluation of existing wireless technology examination of Bluetooth
 - Detailed study of a session protocol, OBEX
 - How should IEEE 1451 transducers communicate on OBEX?
- Develop a VHDL software model of system.
 - Behavioral description of client-side functionality.
 - Simulations

Department of Electrical Engineering

Example of a Smart Transducer Network

Background

- Remotely monitored UAV (Unmanned Aerial Vehicle)
- Applications
 - Military surveillance and law enforcement
 - Environment monitoring and pollution control
 - Media coverage

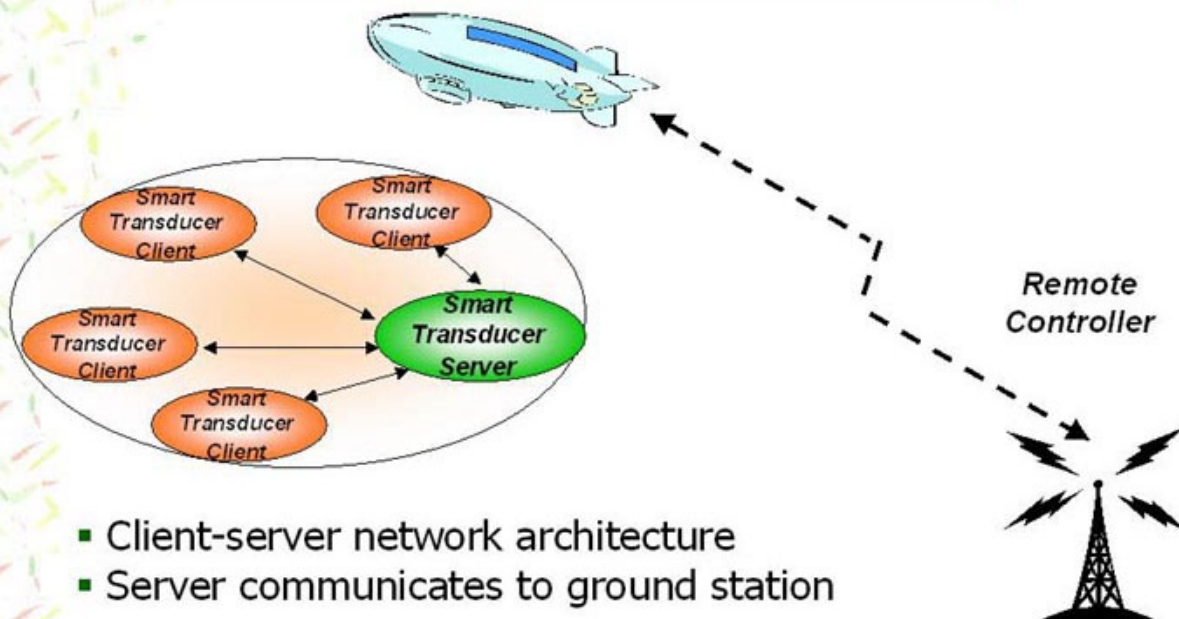


Department of Electrical Engineering



Example of a Smart Transducer Network

Background



- Client-server network architecture
- Server communicates to ground station



Department of Electrical Engineering



Wireless Flying Sensor Platform

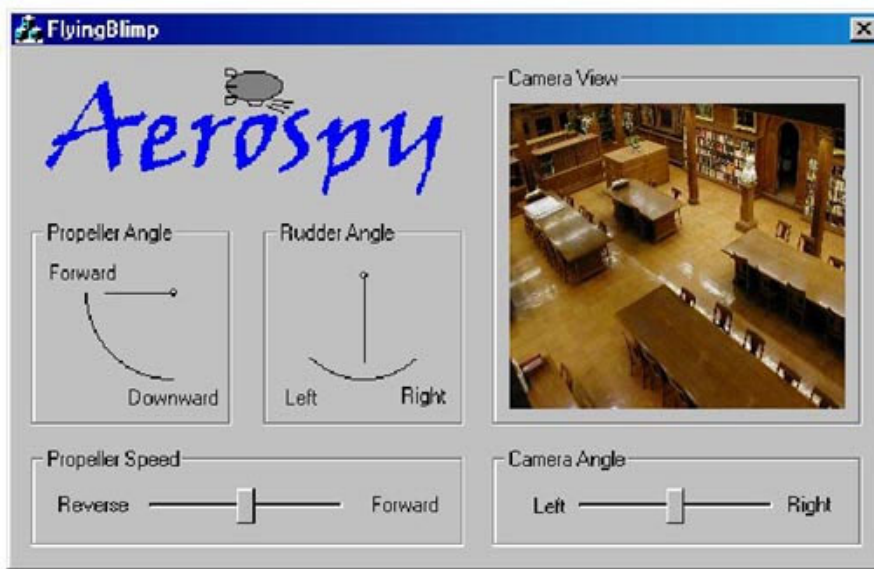
- In the lab and then at the videoconference.



Department of Electrical Engineering



User Interface

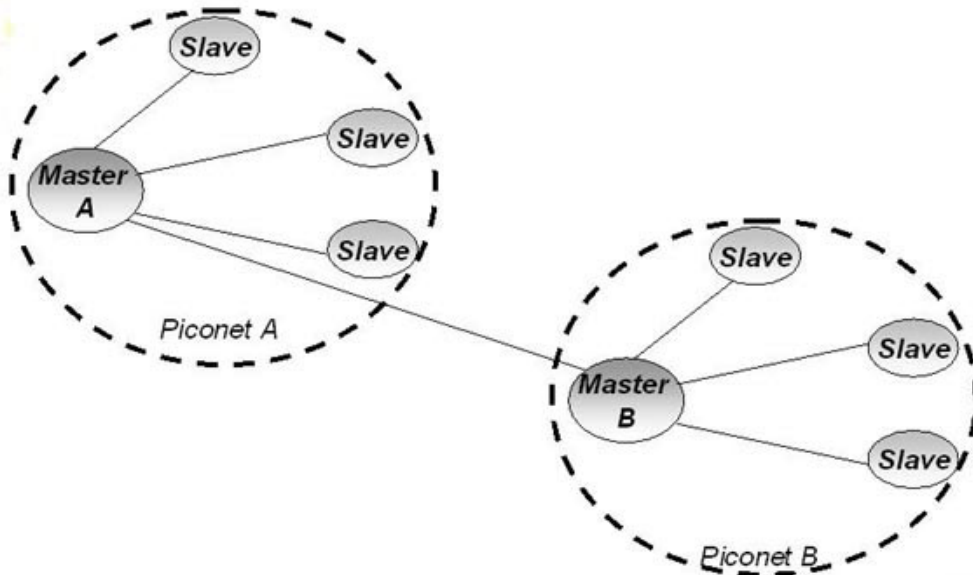


Department of Electrical Engineering



Bluetooth Network Architecture

Background

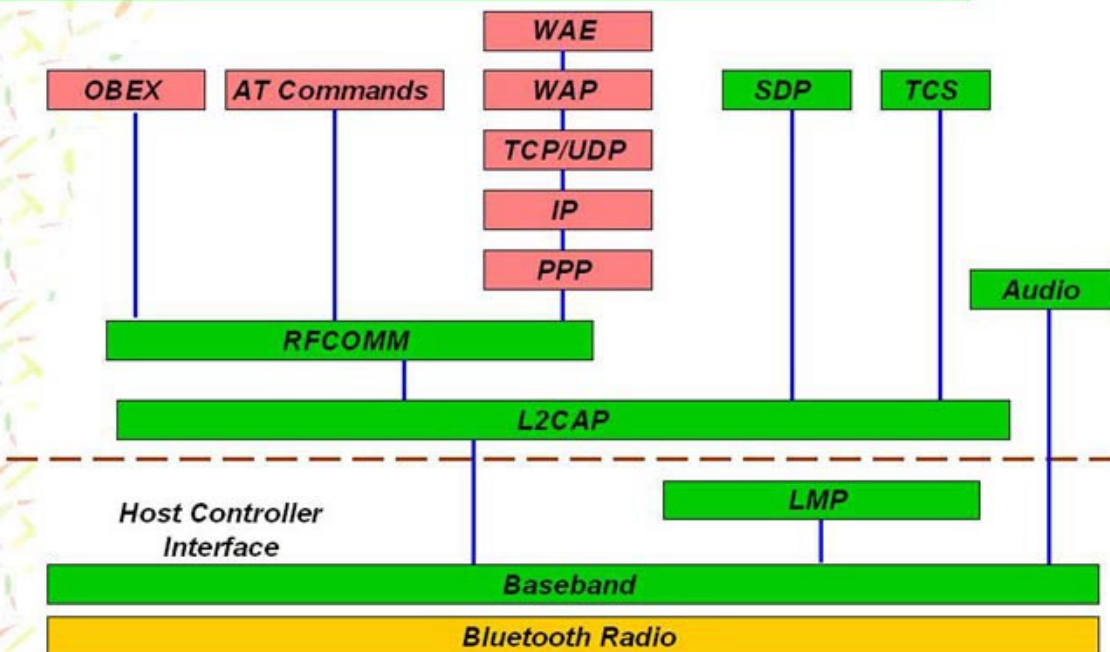


Department of Electrical Engineering



Bluetooth Protocol Stack

Background



Department of Electrical Engineering



Protocol for interfacing Bluetooth with IEEE 1451

Background

- IEEE 1451 Client nodes "Execute" the "Perform" operation on Server nodes
- Session-level protocol required for client server communication
- Two options
 - TCP/IP – Large overheads
 - OBEX – Light version of HTTP
- OBEX is the chosen alternative



Department of Electrical Engineering



Background

OBEX Protocol

- Primarily developed by IrDA and adopted by Bluetooth for interoperability
- Example applications are for short-range business card exchange or synchronization
- Can operate on both RFCOMM as well as TCP/IP as transport
- Defined by
 - OBEX Object Model
 - OBEX Session Protocol

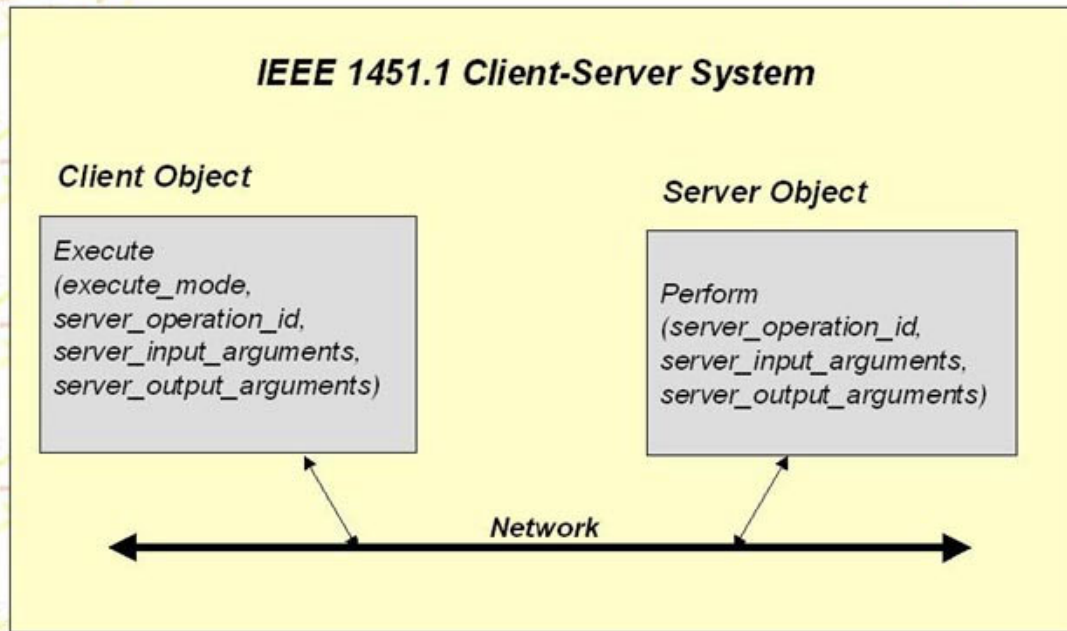


Department of Electrical Engineering



IEEE 1451 Communication on Bluetooth

Design

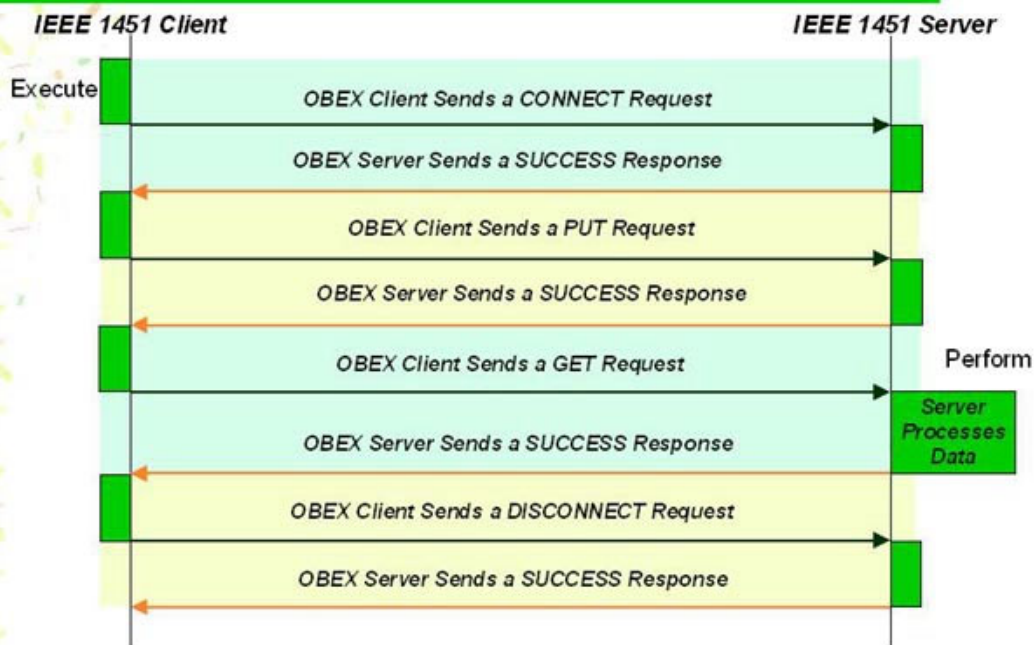


Department of Electrical Engineering



OBEX Session Protocol for Smart Transducers

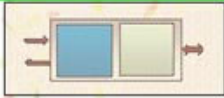
Design



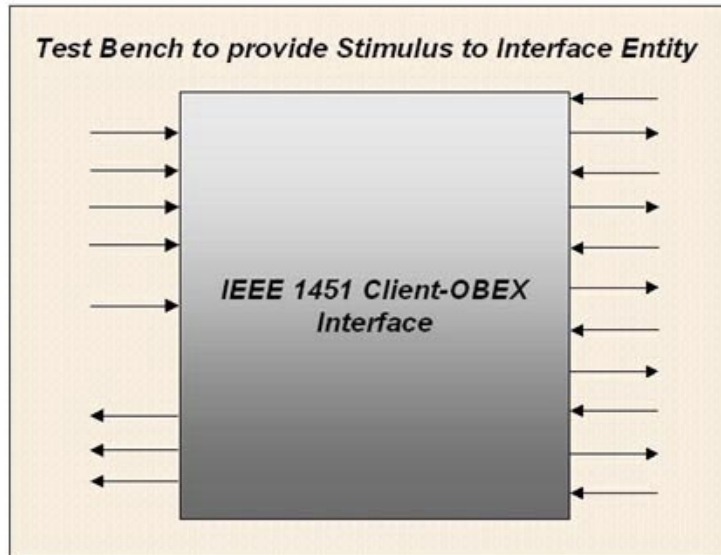
Department of Electrical Engineering



Test Bench for Design



Simulates signals for providing Client requests and receiving Server response with error codes



Simulates signals for providing server responses in OBEX format



Department of Electrical Engineering



VHDL Design of Interface Entity

```

ENTITY interface IS
PORT(
  client_request : IN BIT;
  client_abort : IN BIT;
  serverid : IN UInteger8;
  server_operation : IN INTEGER;
  server_input_args : IN integer_array;
  received_pkt : IN BIT;
  connect_success_pkt : IN byte_array(0 TO 6);
  get_success_pkt : IN byte_array(0 TO payload_packet_length);
  success_pkt : IN byte_array(0 TO 2);
  client_ack : OUT BIT;
  server_output_args : OUT integer_array;
  portcode : OUT UInteger8;
  performcode : OUT UInteger8;
  majorcode : OUT UInteger8;
  minorcode : OUT UInteger8;
  connect_pkt : OUT byte_array(0 TO 10);
  put_pkt : OUT byte_array(0 TO payload_packet_length);
  get_pkt : OUT byte_array(0 TO 2);
  disconnect_pkt : OUT byte_array(0 TO 2);
  abort_pkt : OUT byte_array(0 TO 2);
  session_state : OUT state);
END interface;
  
```



Department of Electrical Engineering



VHDL Design of OBEX Session

```

ENTITY obex_session IS
  PORT(check : IN BIT; -- clock for session protocol
        client_abort : IN BIT; -- client node aborts operation
        server_id : IN UInteger8;
        obex_body_object : IN obex_body;
        obex_send : IN BIT;
        obex_receive : IN BIT;
        connect_success_pkt : IN byte_array(0 TO 6); -- success to connect
        get_success_pkt : IN byte_array(0 TO payload_packet_length); --for output
        success_pkt : IN byte_array(0 TO 2); -- success to put, disconnect, abort
        obex_result_object : OUT obex_result;
        session_state : OUT state; -- state of OBEX session communication
        connect_pkt : OUT byte_array(0 TO 10); -- start obex session with connect
        put_pkt : OUT byte_array(0 TO payload_packet_length); -- put obex body
        get_pkt : OUT byte_array(0 TO 2); -- get results from server
        disconnect_pkt : OUT byte_array(0 TO 2); -- end session
        abort_pkt : OUT byte_array(0 TO 2); -- stop operation in the middle of a session
        timeout_error : OUT BIT -- indicate session has timed out
  );
END obex_session;

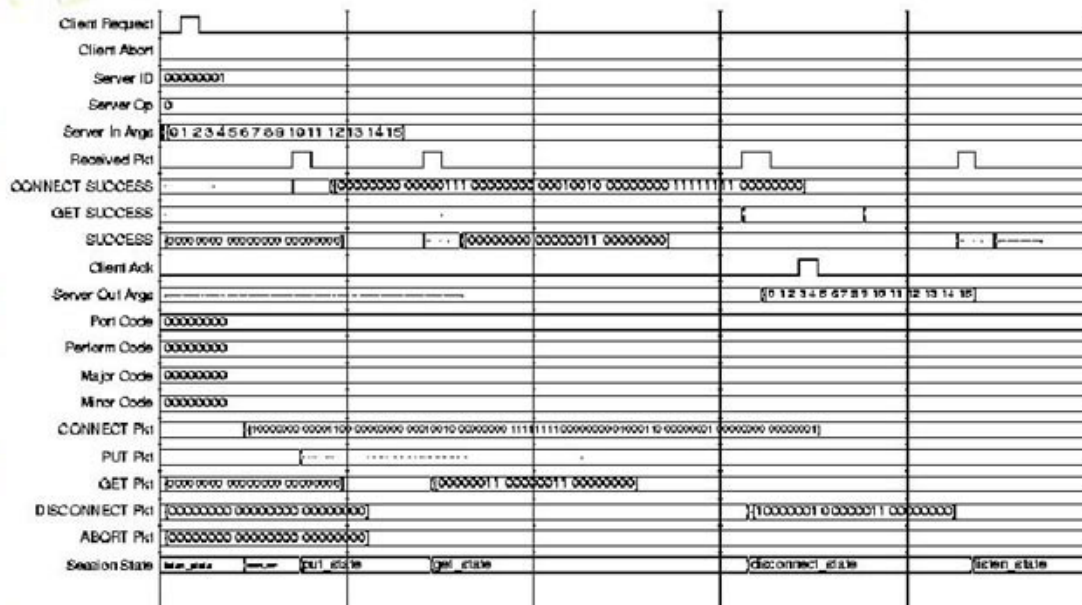
```



Department of Electrical Engineering



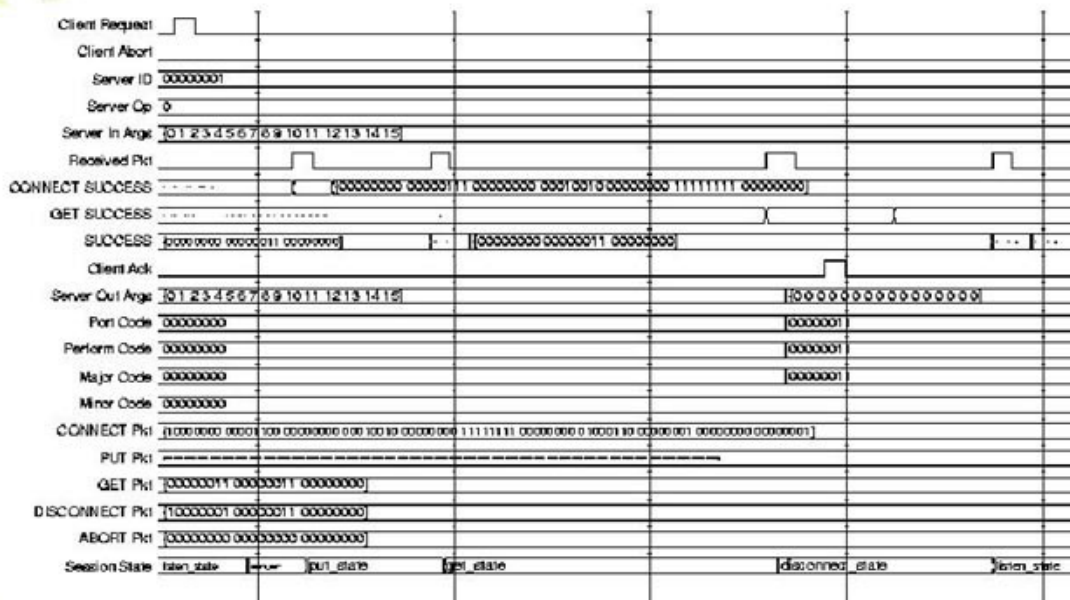
Client-Server VHDL Simulation



Department of Electrical Engineering



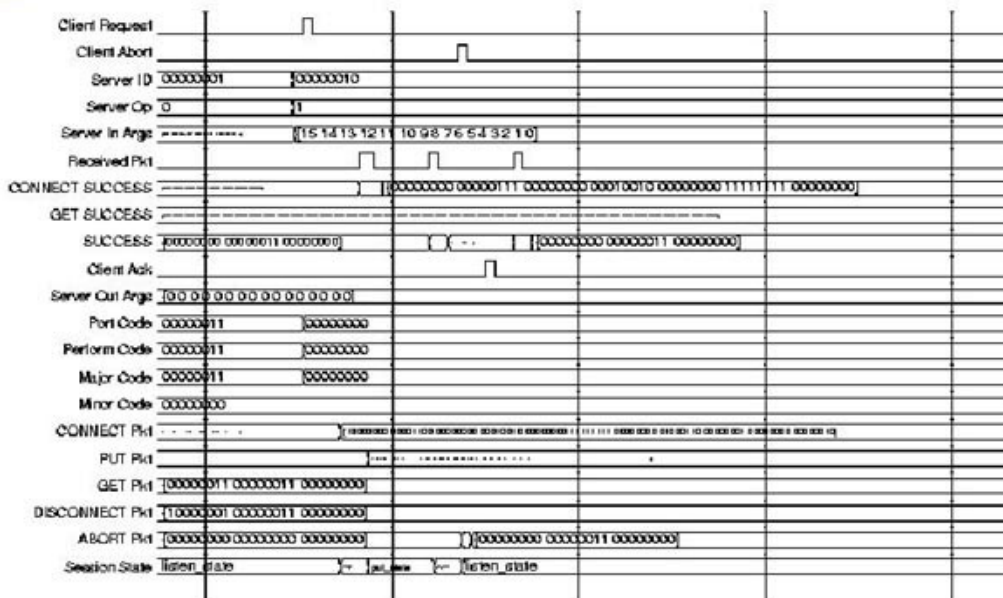
Server Responds with error



Department of Electrical Engineering



Client Abort



Department of Electrical Engineering



Conclusions

- VHDL descriptions enables an implementation-oriented investigation of wireless sensor specifications.
- Easier to assess the feasibility and “cost” of various features by “what if” test bench scenarios.
- Easier to “visualize” complex behavior and catch potential problems.
- Prototyping for iterative specifications.



Department of Electrical Engineering



"Development of Wireless Sensor Technology for Machine Monitoring"

***Dr. Mark F. Bocko
Oceana Sensor Technologies
Virginia Beach, VA***

M. Bocko - Oceana Sensor Technologies - 6/4/01



Presentation Outline

- ***Machine monitoring applications of wireless intelligent sensors***
- ***Wireless intelligent sensor architecture***
- ***Role of IEEE1451 in OST developments***
- ***Summary***

M. Bocko - Oceana Sensor Technologies - 6/4/01

Proposed Approach

*The equipment tells the people when it
will need attention -*

- *Intelligent platforms built from*
- *Intelligent systems built from*
- *Intelligent components*

..... in an open architecture

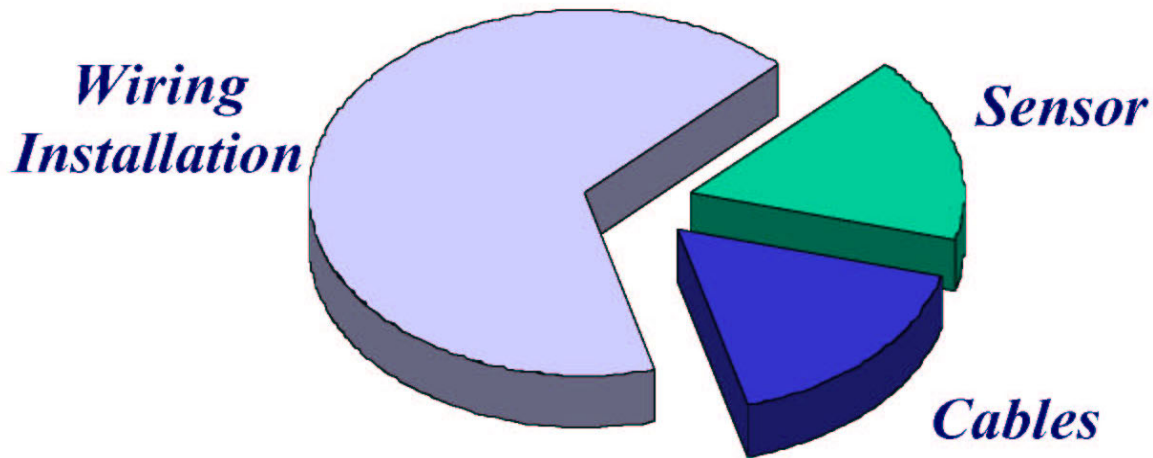
M. Bocko - Oceana Sensor Technologies - 6/4/01

Types of sensing in machine monitoring

- *Vibration*
- *Force*
- *Pressure*
- *Temperature*
- *Electrostatic, magnetic (particulates)*
- *IR spectroscopy (fluid condition)*
- *.....*

M. Bocko - Oceana Sensor Technologies - 6/4/01

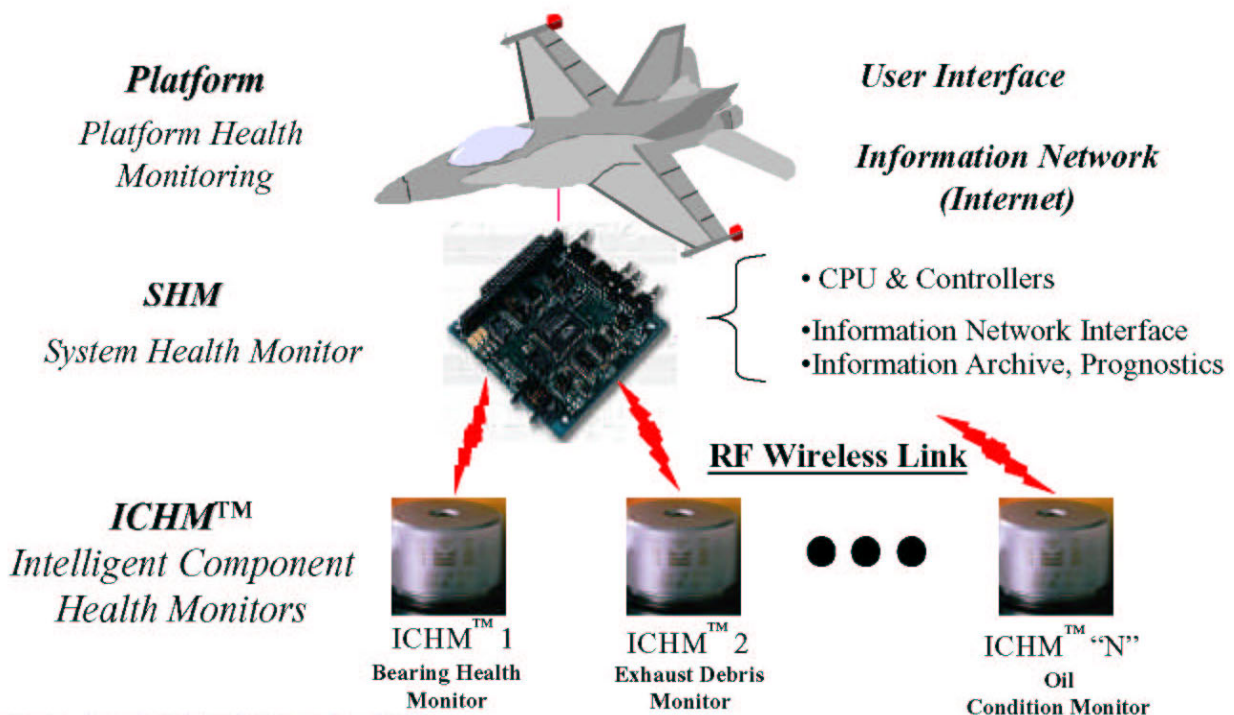
Typical Industrial Instrumentation Costs



Wiring installation costs (including documentation) is limiting installation of needed sensors

M. Bocko - Oceana Sensor Technologies - 6/4/01

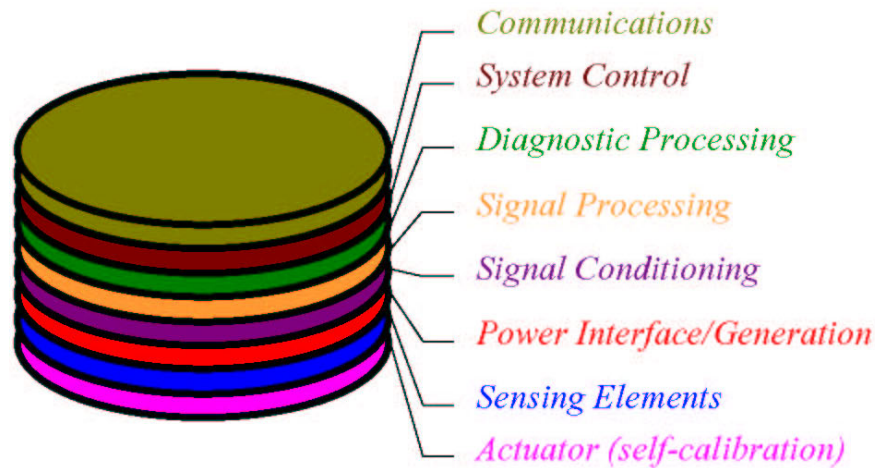
Machine Health Monitoring Application



M. Bocko - Oceana Sensor Technologies - 6/4/01

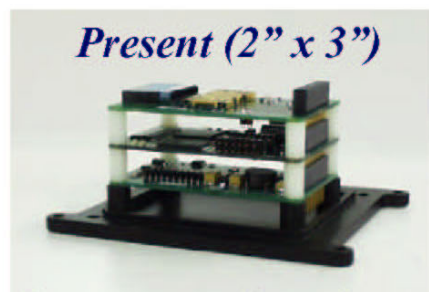
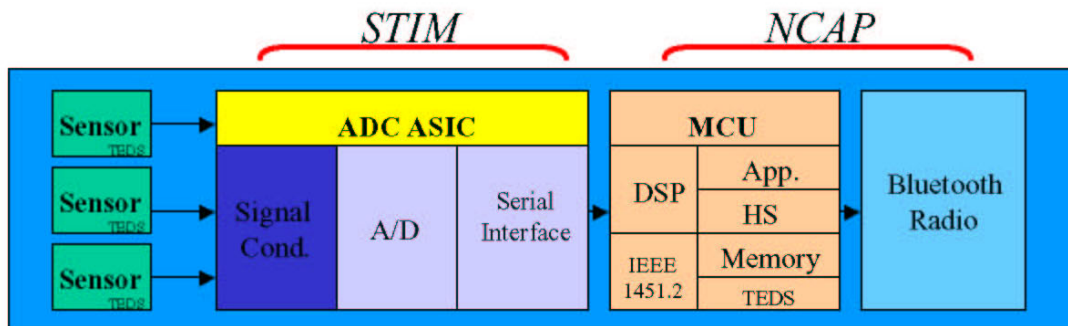
The layers of an ...

“Intelligent Component Health Monitor”



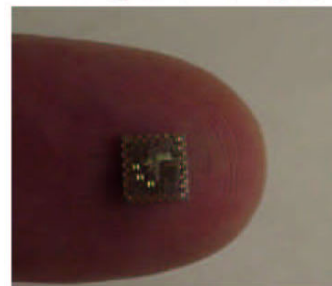
M. Bocko - Oceana Sensor Technologies - 6/4/01

Wireless Intelligent Sensor Architecture



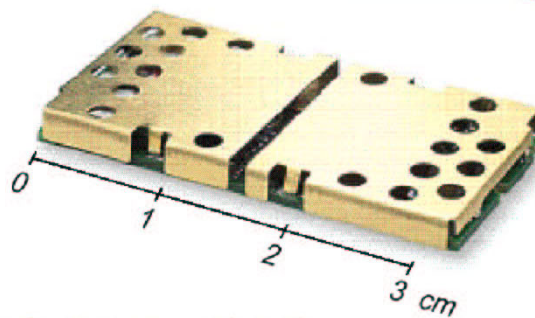
6 sensors, DSP, Bluetooth

Target (2 chips)



M. Bocko - Oceana Sensor Technologies - 6/4/01

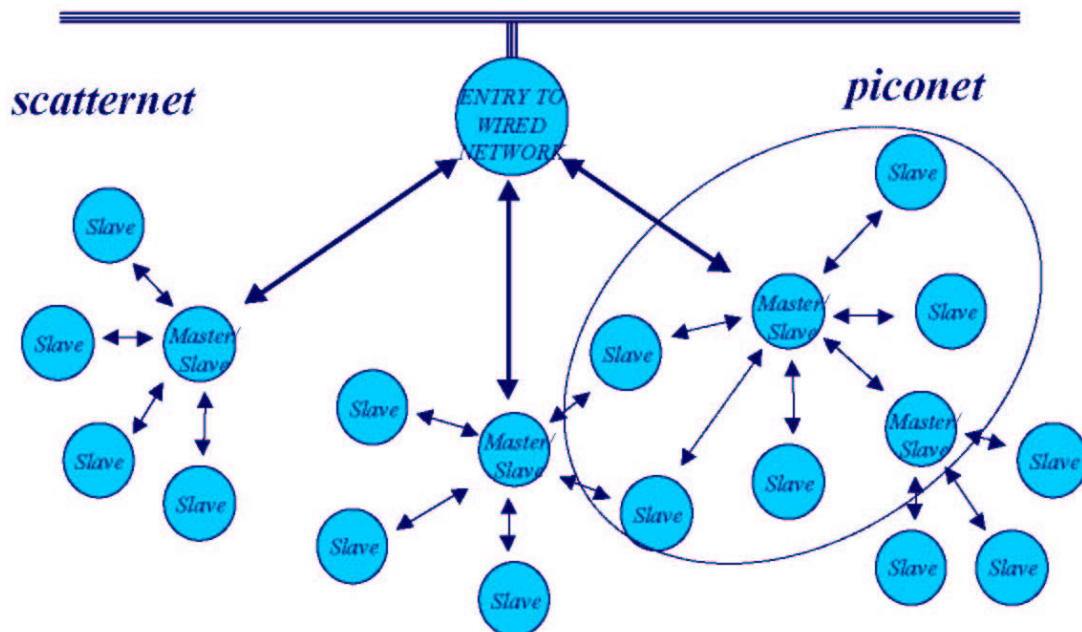
Bluetooth Wireless



- Bluetooth may become de facto standard
- Digital spread spectrum, encrypted data
- Flexible ad hoc networking
- Up to 760 kbit/s
- 1 mW -10 m ; 100 mW - 100m, ISM band
- Will be single chip (CMOS) soon

M. Bocko - Oceana Sensor Technologies - 6/4/01

Bluetooth Networking



M. Bocko - Oceana Sensor Technologies - 6/4/01

Wireless Issues in Industrial Automation

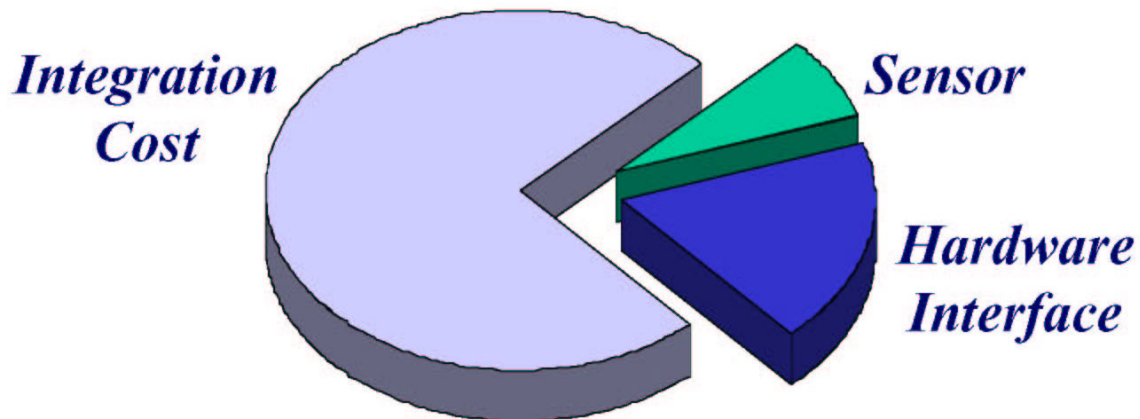
- *Over-the-air Interoperability*
- *Synchronization*
- *Latency*
- *Physical, Electro-Magnetic Environment*
- *Security*

OST chairs a Bluetooth SIG Working Group to address these issues.

M. Bocko - Oceana Sensor Technologies - 6/4/01

!! Pure Conjecture !!

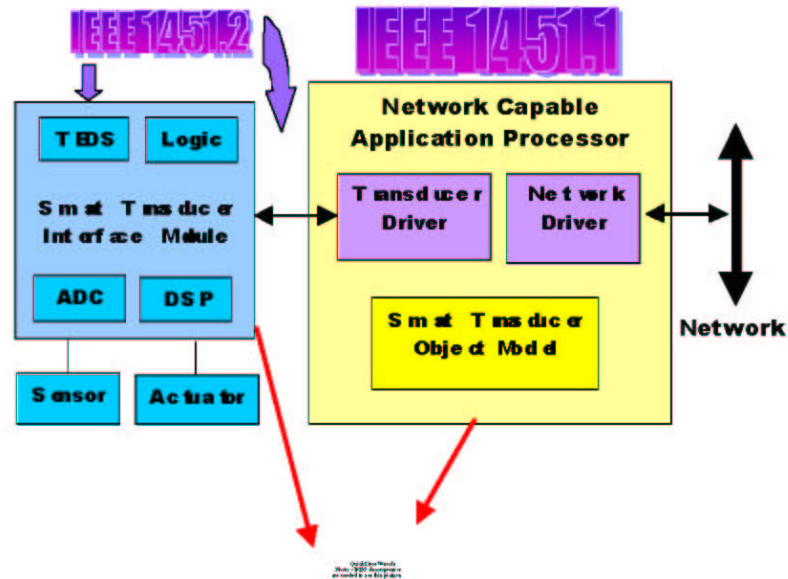
Typical Sensor Integration Costs



Sensor integration costs (including documentation) is limiting installation of needed sensors

M. Bocko - Oceana Sensor Technologies - 6/4/01

IEEE1451 in the OST development plan



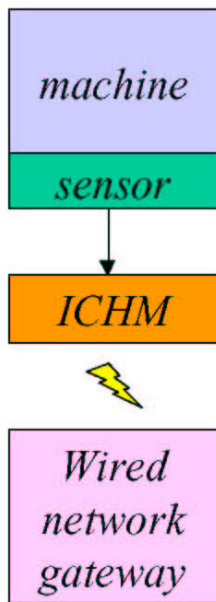
M. Bocko - Oceana Sensor Technologies - 6/4/01

IEEE-1451 Transducer Interface Standard

- **IEEE 1451.1**
 - Defines the Transducer Object Model
- **IEEE 1451.2**
 - Defines the interface for integration of specific smart sensors to microprocessors
 - Physical interface
 - TEDS
 - Protocol
- **IEEE 1451.4**
 - Defines an interface to bring TEDS functionality to legacy sensor systems

M. Bocko - Oceana Sensor Technologies - 6/4/01

Demonstration



www.senseblue.com

QuickTime™ and a
PNG decompressor
are needed to see this picture.

M. Bocko - Oceana Sensor Technologies - 6/4/01

Data carrying wires may be eliminated; can we remove power connections too?

- *Power grid is not always available*
- *Derive power from environment?*
 - *Vibration*
 - *Magnetic induction (rotating machines)*
 - *Solar*
 - *Thermal Energy Scavenging (hot machines)*

M. Bocko - Oceana Sensor Technologies - 6/4/01

“Plug and Play” → “Play”

- *Key elements of strategy*
 - *Wireless communication*
 - *Open interoperable standards*
 - *IEEE1451*
 - *Bluetooth wireless*
 - *Power scavenging*

IV. List of Attendees

Hunter Babcock
Graham-White Mfg.
1242 Colorado St.
Salem, VA 24153
540-387-5600

Gregory A. Balchin
B35, NSWCD
17320 Dahlgren Rd.
Dahlgren, VA 22448
540-653-6837

Tom Baudendistel
Delphi Automotive
P.O. Box 1245
Dayton, OH 45401
thomas.a.baudendistel@delphiauto.com
937-455-6660

Steven Bibyk
Ohio State University
Columbus, OH
bibyk.1@osu.edu
614-292-1300

Brian Bischoff
Red Wing Technologies
115 Hennepin Ave
Minneapolis, MN
215-527-1986

Mark F. Bocko
Univ. of Rochester, ECE Dept.
205 Hopeman Hall
Rochester, NY 14627
bocko@oceanasensor.com
716-275-4879

Dan Boyd
IZEX Technologies
5945 Golden Valley Rd
Golden Valley, MN 55422
dan.boyd@izex.com
763-544-1001

Richard Bragg
3M
3M Center Bldg. 270-4-N-03
St. Paul, MN 55144-1000
rcbragg@mmm.com
651-733-0118

Adam Brody
Accentrue
3773 Willow Rd
Northbrook, IL 60062
adam.b.brody@accentrue.com
847-714-3747

Bill Brotherton
Sensotec
2080 Arlingate Lane
Columbus, OH 43228
bbrotherton@sensotec.com
614-850-6000

John T. Cain
Dexter Electronic Materials
110 Brookfield Lane
Geneva, IL 60134
john.cain@loctite.com
630-208-0547

Christopher Corrado
Greene, Tweed & Co.
2075 Detwiler Rd.
Kulpsville, PA 19443
ccorrado@gtweed.com

Eugene Cullie
Accusonic Technologies
25 Bernard St. Jean Drive
E. Falmouth, MA 02536
g.cullie@accusonic.com

Krishna Davarasetty
Goodrich Corp.
5353 52nd St. S.E.
Grand Rapids, MI 49588
kris_davarasetty@avionics.bfg.com
616-285-4251

Jos Dominguez
Accusonic Technologies
25 Bernard St. Jean Drive
E. Falmouth, MA 02536
g.cullie@accusonic.com

Thomas Doney
Nestle PTC New Milford
201 Housatonic Ave
New Milford, CT 06776
thomas.doney@rdct.nestle.com
860-355-6304

Diana Dulla
Co-op Publishing
P.O. Box 426192
Cambridge, MA 02142
617-465-7454

Prabal Dutta
NetEnabled
1275 Kinnear Rd.
Columbus, OH 43212
prabal@netenabled.com

Lars Enevoldsen
Grundfos
Poul D. Jensens Bej 7
DK-8850, Bjerringbro
Denmark
lenevoldsen@grundfos.com

Mark Farrar
Westinghouse SRC
Bldg. 723-A
Aiken, SC 29841
mark.farrar@srs.gov
803-725-1786

Otto Fest
Otek Corp.
4016 E. Tennessee St.
Tucson, AZ 85714
drfest@otekcorp.com
520-748-7900

Kevin Finkbiner
Teckko
2001 Fulling Mill Rd.
Middletown, PA 17057
kfinkbiner@actpower.com
717-939-2300

Leslie Fowler
Lord Corporation
P.O. Box 8012
Cary, NC 27512-8012
leslie.fowler@lord.com
919-469-3443

Ron Fredricks
2046 Foxboro N.W.
Grand Rapids, MI 49504
ron_fredricks@ameritech.net
616-741-9134

Robert Fricke
ADL
Acorn Park
Cambridge, MA 02140
fricke.robert@adlittle.com
617-498-5180

David Fries
U. South Florida
140 7th Ave So
St. Petersburg, FL 33701
dfries@marine.usf.edu
727-553-3961

Fernando GenKuong
Endevco
30700 Rancho Viejo Rd
San Juan Capistrano, CA 92675
fernando@endevco.com
949-493-8181

James Gilsinn
NIST
100 Bureau Drive, Stop 8230
Gaithersburg, MD 20899
james.gilsinn@nist.gov
301-975-3865

Paul H. Gusciora
Chevron Research & Technology
100 Chevron Way #51-1202
Richmond, CA 94802-0627
phgu@chevron.com
510-242-2972

Ed Herceg
Macrosensors
7300 U.S. Rt. 130 North
Pennsauken, NJ 08110
eeh@macrosensors.com
856-662-8000

Gerard Hill
Axon
2021 Lakeshore Drive
New Orleans, LA 70122
gerhill@axon.com
504-282-8119

Hui Huang
NIST
100 Bureau Drive, Stop 8230
Gaithersburg, MD 20899
huang@cme.nist.gov
301-975-3427

Todd James
James Consulting
2713 Skelton Lane
Blacklick, OH 43004
tjames@mindspring.com
614-855-2741

Robert N. Johnson
Telemonitor Inc.
9055F Guilford Road
Columbia, MD 21046
robertj@telemonitor.com
410-312-6621
Muammer Kec
UWM
P.O. Box 71035
Milwaukee, WI 53211
kec@uwm.edu

David Keenan
Advanced MicroMachines,
Goodrich Corp.
14300 Judicial Road, FF--14
Bunsville, MN 55306
david.keenana@asd.bfg.com
952-892-4852

Chung Lee
GMC
9301 W. 55th
La Grange, IL 60525
chung.shiklee@gm.com
708-387-5606

Jong-Duke Lee
KINSCO
372-8, Seokyo-dong, Mapo-Ku
Seoul, Korea 121-210
kinsco@kebi.com
822-338-9667

Kang Lee
NIST
100 Bureau Drive, Stop 8220
Gaithersburg, MD 20899
kang.lee@nist.gov
301-975-6604

Sinyu Li
275 Fitzpatrick Hall
Notre Dame, IN 46556
xli2@nd.edu
219-631-6916

Xinyu Li
275 Fitzpatrick Hall
Notre Dame, IN 46556
xli2@nd.edu

Wayne MacKenzie
BWX Technologies
P.O. Box 785
Lynchburg, VA 24505
804-522-6053

Gary Manninen
IZEX Technologies, Inc.
5945 Golden Valley Road
Golden Valley, MN 55422
gary.manninen@izex.com

Brent McAdams
USTC
10713 N. Oak Hills Pkwy. St. A
Baton Rouge, LA 70810
bmcadams@ustelemetry.com
225-761-9501

Chris McGee
Lake Shore
574 McCorkle Blvd.
Westerville, OH 43082
cmcgee@lakeshore.com
614-891-2243

Duncan McIver
12050 Jefferson Ave., Suite 350
Newport News, VA 23606
757-269-0025

Pedro Medelius
DNX-10
Kennedy Space Center, FL 32899
medelius@att.net
321-867-6335

Mike Moore
Oak Ridge National Lab.
Bldg. 3500, MS-6006, P.O. Box 2008
Oak Ridge, TN 37831-6006
mooremr@ornl.gov

David Nagel
GWU
2033 K St. NW (340J)
Washington, DC 20052
nagel@seas.gwu.edu
202-994-5293

Mike Nase
210 W. Stone Ave
Greenville, SC 29609
mnase10566@aol.com
864-241-0986

Dieter Nowak
Micron Corp.
158 Orchard Ln.
Winchester, TN 37398
dieternowak@yahoo.com
931-967-9859

S Orlove
PRSLMC
5311 N. Kimball Ave
Chicago, IL 60625-4713

Mohan Pawar
Mio Instruments LLC
1110 Midway Rd. #172
Menasha, WI 54952

Heins K. Pedersen
Grundfos
P.D. Jensens VEJ7
DK-8850, Denmark
hkpedersen@grundfos.com

Jianhong Pei
University of Notre Dame
Notre Dame, IN 46556
jpei@nd.edu
219-631-9027

David Potter
National Instruments
11500 N. Mopac Expy
Austin, TX 78759
david.potter@ni.com
512-683-5489

David Powers
Engineering Plus
1129 West Lunt Ave
dpowers@220221.com
847-923-0300

John Ramthun
3M Center
Bldg. 518-01-01
St. Paul, MN 55144-1000
jaramthun@mmm.com

Bob Randall
YSI Mass
13 Atlantis Drive
Marion, MA 02738
brandall@ysi.com
508-748-0366

John Randazzo
Dynacs Co.
DNX-28
Kenedy Space Center, FL 32899
john.randazzo-1@ksc.nasa.gov
321-867-6934

George Rawa
Greene, Tweed & Co
2075 Detwiler Rd.
Kulpsville, PA 19443
grawa@gtweed.com

Manuel Ruiz-Sandoval
Univ. of Notre Dame
Dept of Civil Engineering
Notre Dame, IN 46556

Frank Salgado
Point Six
391 Codell Dr.
Lexington, KY 40509
frankie@pointsix.com
859-266-3606

Steven F. Sciamanna
Chevron Research & Technology
100 Chevron Way #51-1202
Richmond, CA 94802-0627
sfsc@chevron.com
510-242-5075

David Scott
Whirlpool
750 Monte Rd.
Benton Harbor, MI 49022
david.e.scott@email.whirlpool.com
616-923-3858

Del Singleton
Bourns Inc.
1200 Columbia Ave.
Riverside, CA 92507

Viktor Slobotyan
Apprise Technologies
4802 Oneta Street
Duluth, MN 55807
vsloboty@appristech.com
218-624-2800

Stan Smith
Kaman Instrumentation
3450 Nevada Ave
smiths-kac@kaman.com
719-635-5936

Steve Smith
Oak Ridge National Lab
Bldg. 3500, MS-6006, P.O. Box 2008
Oak Ridge, TN 37831-6006
smithsf@ornl.gov
865-576-2184

Gregory Steinthal
Cyrano Sciences
73 N. Vinedo
Pasadena, CA 91107
steinthal@cyranosciences.com

Gerhard Stezzer
Austria Mikro Systeme Int.
A-8141 Urtemprentotter
Austria
00114331365005473

Jim Tatera
Tatera & Assoc.
2038 Ridgewood Dr.
Madison, IN 47250
jtatera@seidata.com
812-215-2301

Michael Thomas
Equitech Diag.
37 Mt. Pleasant Ave.
East Hanover, NJ 07936
info@equitechdiag.com
973-887-1660

Lou Trippel
YSI, Inc.
1725 Brannum Lane
Yellow Springs, OH 45387
ltrippel@ysi.com
937-767-7241

Karen Vardanyan
SEUA
12/2,36, Yegh Tadevosyan
Yerevan, Armenia
karenv@undp.am

37 41 42 336 556 6073

Paul Viggium
Equitech Diag.
37 Mt. Pleasant Ave.
E. Hanover, NJ 07936
doctorpaul@netzero.net
973-887-1660

Alex Wang
N018 Jalan ISHAK
Singapore, 419347

Victor Weiss
376 Waycliffe Cir.
Wayzata, MN 55391

Chuck Whiting
Analog Devices
804 Woburn St.
Wilmington, MA 01887
charles.whiting@analog.com
781-937-1540

Ed Whittaker
Stevens Institute of Tech.
Dept. of Physics
Hoboken, NJ 07030
ewhittak@stevens-tech.edu
201-216-5707

Ed Zdankiewicz
Hygrometrix
1128 W. Pleasant Valley Rd.
Cleveland, OH 44134
edzdankiewicz@att.net
216-374-3377

Ken Zemach
Exponent
149 Commonwealth Dr.
Menlo Park, CA 29841

Allen Ziaimehr
Augustine Medical
10393 West 70th St.
Eden Prairie, MN 55344
hziaimeh@augmed.com
952-947-1263